Table P.1/H.263 – Fill-mode bits/action

Fill-mode bits	Fill action
0 0	*color*
0 1	*black*
1 0	*gray*
1 1	*clip*

If the fill mode is *clip*, the coordinates of locations in the prior reference picture are independently limited as in the Unrestricted Motion Vector mode so that pixel values outside of the prior reference picture area are estimated by extrapolation from the values of pixels at the image border. If the fill mode is *black*, luminance samples outside of the prior reference picture area are assigned a value of $Y = 16$ and chrominance samples are assigned a value of $C_B = C_R = 128$. If the fill mode is *gray*, luminance and chrominance values are assigned a value of $Y = C_B = C_R = 128$. If the fill mode is *color*, then additional fields are sent to specify a fill color, as described in the next subclause.

P.2.4 Fill Color Specification (Y_FILL, C_B_EPB, C_B_FILL, C_R_EPB, C_R_FILL) (26 bits)

If the fill mode is *color* and the picture is not an EP-picture, then the fill-mode bits are followed in the bitstream by three eight-bit integers, *Y_fill*, C_B*_ fill*, and C_R*_ fill*, which specify a fill color precisely. Between these three eight-bit integers are two emulation prevention bits (C_B_EPB and C_R_EPB) each of which is equal to 1. The format of this color specification, which is present only when the fill mode is *color*, is shown in Figure P.2. Each eight-bit integer field is sent using its natural representation. For an EP-picture, the fill-mode action (and fill color) is the same as that for the reference layer, and the fill color specification is not sent.



Figure P.2/H.263 – Format of fill mode and fill color specification data

P.3 Resampling algorithm

The method described in this clause shall be mathematically equal in result to that used to generate the samples of the resampled reference picture. Using the integer warping parameters $w_x^0, w_y^0, w_x^x, w_y^x, w_x^y, w_y^y, w_x^{xy}$, and w_y^{xy} integer parameters u_x^{00}, u_y^{00}, u_x^{H0}, u_y^{H0}, u_x^{0V}, u_y^{0V}, u_x^{HV}, and u_y^{HV} which denote the x- and y-displacements at the corners of the luminance field in $\frac{1}{32}$-pixel accuracy (the actual displacements are obtained by dividing these values by 32) are defined as:

$$u_x^{00} = 16 w_x^0 \qquad u_y^{00} = 16 w_y^0$$

$$u_x^{H0} = 16\left(w_x^0 + w_x^x + 2(H_R - H)\right) \qquad u_y^{H0} = 16\left(w_y^0 + w_y^x\right)$$

$$u_x^{0V} = 16\left(w_x^0 + w_x^y\right) \qquad u_y^{0V} = 16\left(w_y^0 + w_y^y + 2(V_R - V)\right)$$

$$u_x^{HV} = 16\left(w_x^0 + w_x^x + w_x^y + w_x^{xy} + 2(H_R - H)\right) \qquad u_y^{HV} = 16\left(w_y^0 + w_y^x + w_y^y + w_y^{xy} + 2(V_R - V)\right)$$

Next, H' and V', which denote the horizontal and vertical size of the *virtual frame*, are defined as the smallest integers that satisfy the following condition:

$$H' \geq H, V' \geq V, H' = 2^m, V' = 2^n, m \text{ and } n \text{ are positive integers}$$

By applying bilinear extrapolation to the corner vectors of the luminance field, the integer parameters u_x^{LT}, u_y^{LT}, u_x^{RT}, u_y^{RT}, u_x^{LB}, u_y^{LB}, u_x^{RB}, and u_y^{RB} which denote the *x*- and *y*-displacements of the luminance field at the *virtual points* $(x, y) = (0, 0)$, $(H', 0)$, $(0, V')$, and (H', V') in $\frac{1}{32}$-pixel accuracy (the actual displacements are obtained by dividing these values by 32) are defined as:

$$u_x^{LT} = u_x^{00} \qquad u_y^{LT} = u_y^{00}$$

$$u_x^{RT} = \left((H - H')u_x^{00} + H' u_x^{H0}\right) // H \qquad u_y^{RT} = \left((H - H')u_y^{00} + H' u_y^{H0}\right) // H$$

$$u_x^{LB} = \left((V - V')u_x^{00} + V' u_x^{0V}\right) // V \qquad u_y^{LB} = \left((V - V')u_y^{00} + V' u_y^{0V}\right) // V$$

$$u_x^{RB} = \left((V - V')\left((H - H')u_x^{00} + H' u_x^{H0}\right) + V'\left((H - H')u_x^{0V} + H' u_x^{HV}\right)\right) // (HV)$$

$$u_y^{RB} = \left((V - V')\left((H - H')u_y^{00} + H' u_y^{H0}\right) + V'\left((H - H')u_y^{0V} + H' u_y^{HV}\right)\right) // (HV)$$

where "//" denotes integer division that rounds the quotient to the nearest integer, and rounds half integer values away from 0.

In the remainder of this annex, it is assumed that the centres of the pixels lie at the points $(x, y) = \left(i + \frac{1}{2}, j + \frac{1}{2}\right)$ for both the luminance and chrominance fields. Integer parameters *i, j* are defined as:

- $i = 0, \ldots, H - 1$ and $j = 0, \ldots, V - 1$ for luminance; and
- $i = 0, \ldots, \frac{H}{2} - 1$ and $j = 0, \ldots, \frac{V}{2} - 1$ for chrominance.

This implies that different coordinate systems are used for luminance and chrominance as shown in Figure P.3. Using the coordinate system for chrominance, the integer parameters u_x^{LT}, u_y^{LT}, u_x^{RT}, u_y^{RT}, u_x^{LB}, u_y^{LB}, u_x^{RB}, and u_y^{RB} defined above can also be regarded as the *x*- and *y*-displacements of the chrominance field at virtual points $(x, y) = (0, 0)$, $(H'/2, 0)$, $(0, V'/2)$, and $(H'/2, V'/2)$ in $\frac{1}{64}$-pixel accuracy (the actual displacements are obtained by dividing these values by 64). Using these parameters and an additional parameter *S*, which is defined as 2 for luminance and as 1 for chrominance, the resampling algorithm for the luminance and chrominance fields are defined using common equations.

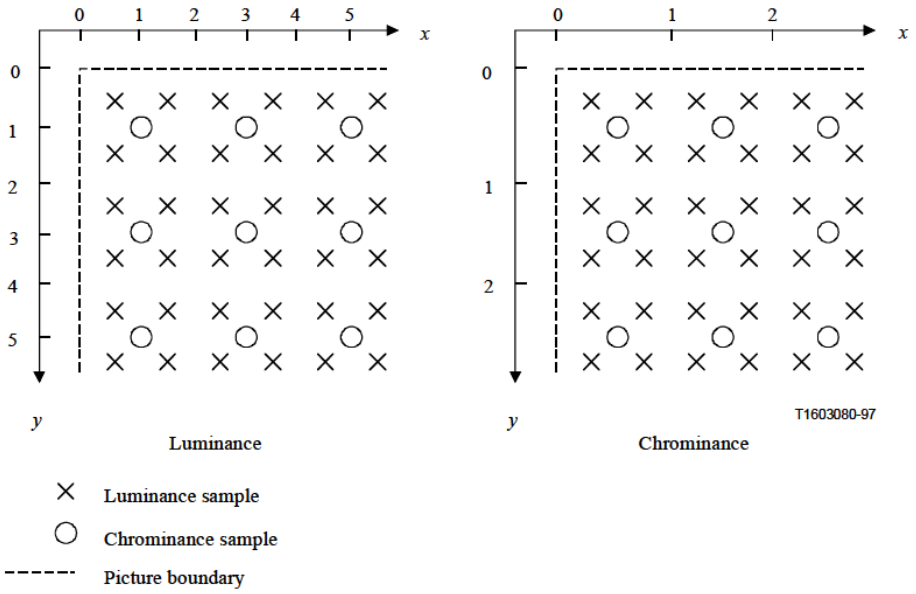


Figure P.3/H.263 – Coordinate systems for luminance and chrominance fields

The integer parameters $u_x^L(j)$, $u_y^L(j)$, $u_x^R(j)$, and $u_y^R(j)$ which denote the x- and y-displacements of the picture field at $(x, y) = (0, j+\frac{1}{2})$ and $(SH'/2, j+\frac{1}{2})$ in $\frac{s}{64}$-pixel accuracy (the actual displacements are obtained by dividing these values by $64/S$) are defined using one dimensional linear interpolation as:

$$u_x^L(j)=\left((SV'-2j-1)u_x^{LT}+(2j+1)u_x^{LB}\right)//(SV') \quad u_y^L(j)=\left((SV'-2j-1)u_y^{LT}+(2j+1)u_y^{LB}\right)//(SV')$$

$$u_x^R(j)=\left((SV'-2j-1)u_x^{RT}+(2j+1)u_x^{RB}\right)//(SV') \quad u_y^R(j)=\left((SV'-2j-1)u_y^{RT}+(2j+1)u_y^{RB}\right)//(SV')$$

where "//" denotes integer division that rounds the quotient to the nearest integer, and rounds half integer values away from 0.

Finally, the parameters that specify the transformed position in the reference picture become:

$$I_R(i,j)=Pi+\left((SH'-2i-1)u_x^L(j)+(2i+1)u_x^R(j)+32H'/P\right)///(64H'/P)$$

$$J_R(i,j)=Pj+\left((SH'-2i-1)u_y^L(j)+(2i+1)u_y^R(j)+32H'/P\right)///(64H'/P)$$

$$i_R(i,j)=I_R(i,j)///P \qquad j_R(i,j)=J_R(i,j)///P$$

$$\varnothing_x=I_R(i,j)-\left(I_R(i,j)///P\right)P \qquad \varnothing_y=J_R(i,j)-\left(J_R(i,j)///P\right)P$$

where:

"///" integer division with truncation towards the negative infinity;

"/" integer division (in this case resulting in no loss of accuracy);

P accuracy of x- and y-displacements (P = 2 when WDA = "10" and P = 16 when WDA = "11" or is absent, see P.2.1 for the definition of WDA);

$\left(\frac{I_R(i,j)}{P}+\frac{1}{2},\frac{J_R(i,j)}{P}+\frac{1}{2}\right)$ (x, y) location of the transformed position (both $I_R(i,j)$ and $J_R(i,j)$ are integers);

$\left(i_R(i,j)+\frac{1}{2},j_R(i,j)+\frac{1}{2}\right)$ (x, y) location of the sampling point near the transformed position (both $i_R(i,j)$ and $j_R(i,j)$ are integers);

$(\varnothing_x,\varnothing_y)$ bilinear interpolation coefficients of the transformed position (both $\varnothing_x$ and $\varnothing_y$ are integers).

The computation of this equation can be simplified by replacing the divisions by shift operations, since $64H'/P = 2^{m+2}$ when $P = 16$ and $64H'/P = 2^{m+5}$ when $P = 2$.

Using these parameters, the sample value, $E_P(i, j)$, of the pixel located at $(x,y)=\left(i+\frac{1}{2},j+\frac{1}{2}\right)$ in the resampled picture is obtained using bilinear interpolation by:

$$E_P(i,j)=\Big((P-\varnothing_y)\big((P-\varnothing_x)E_R(i_R,j_R)+\varnothing_x E_R(i_R+1,j_R)\big)$$

$$+\varnothing_y\big((P-\varnothing_x)E_R(i_R,j_R+1)+\varnothing_x E_R(i_R+1,j_R+1)\big)+P^2/2-1+RCRPR\Big)/P^2$$

where "/" denotes division by truncation. i_R and j_R are simplified notations for $i_R(i,j)$ and $j_R(i,j)$, and $E_R(i_R,j_R)$ denotes the sample value of the pixel located at $(x,y)=\left(i_R+\frac{1}{2},j_R+\frac{1}{2}\right)$ in the reference picture after extrapolation using the proper fill mode if necessary. The value of parameter RCRPR is defined as follows:

- For a B-picture (or the B-part of an Improved PB-frame) which has a P-picture as its temporally subsequent anchor picture, RCRPR is equal to the rounding type (RTYPE) bit in MPPTYPE (see 5.1.4.3) of this temporally subsequent P-picture. This implies that for Improved PB-frames, RCRPR has the same value for the P-part and the B-part.
- For other types of pictures, RCRPR is equal to the RTYPE bit of the current picture.

P.4 Example of implementation

In this clause, an implementation example of the algorithm described in the previous clause is provided as pseudo-code.

P.4.1 Displacements of virtual points

When a large picture is coded, a straightforward implementation of the equation for obtaining parameters u_x^{RB} and u_y^{RB} shown in P.3 may necessitate the usage of variables requiring more that 32 bits for their binary representation. For systems which cannot easily use 64-bit integer or floating point registers, an example of an algorithm which does not require variables with more than 32 bits for the calculation of u_x^{RB} and u_y^{RB} is shown below.

Since H, V, H', and V' are divisible by 4, the definition of u_x^{RB} can be rewritten as:

$$u_x^{RB} = \Big((V_Q - V_Q')\big((H_Q - H_Q')u_x^{00} + H_Q' u_x^{H0}\big) + V_Q'\big((H_Q - H_Q')u_x^{0V} + H_Q' u_x^{HV}\big)\Big) // A$$

where H_Q=H/4, V_Q=V/4, H_Q'=H'/4, V_Q'=V'/4, $A = H_Q V_Q$, and "//" denotes integer division that rounds the quotient to the nearest integer, and rounds half integer values away from 0. Next, parameters T_T and T_B are defined as:

$$T_T = (H_Q - H_Q')u_x^{00} + H_Q' u_x^{H0}$$
$$T_B = (H_Q - H_Q')u_x^{0V} + H_Q' u_x^{HV}$$

for simplicity of description. Using operator "///" which denotes integer division with truncation towards the negative infinity, and operator "%" which is defined as a % b = a – (a /// b) b, the value of u_x^{RB} can be obtained via the following pseudo-code:

```
q = (V_Q - V_Q')*(T_T /// A) + V_Q'*(T_B /// A) + ((V_Q - V_Q')*(T_T % A) + V_Q'
    *(T_B % A)) /// A;
r = ((V_Q - V_Q')*(T_T % A) + V_Q'*(T_B % A)) % A;
if (q < 0)
  u_x^RB = q+(r+(A-1)/2)/A;
else
  u_x^RB = q+(r+A/2)/A;
```

The value of u_y^{RB} can also be calculated using this algorithm.

P.4.2 Resampling algorithm

To simplify the description of the algorithm, a function `prior_sample` is defined. Its purpose is to generate a pixel value for any integer location (i_p, j_p) relative to the prior reference picture sampling grid:

```
    clip(x_min, x, x_max) {
      if (x < x_min) {
        return x_min;
      } else if (x > x_max) {
        return x_max;
      } else {
        return x;
      }
    }
prior_sample (i_p, j_p) {
    if (FILL_MODE = clip) {
        i_c = clip (0, i_p, S*H_R /2-1);
        j_c = clip (0, j_p, S*V_R /2-1);
        return prior_ref[i_c, j_c];
    } else {
        if ((i_p < 0) OR (i_p > S*H_R /2-1) or (j_p < 0) OR (j_p > S*V_R /2-1) {
            return fill_value;
        } else {
            return prior_ref[i_p,j_p];
        }
    }
}
```

In the pseudo-code, `prior_ref[i, j]` indicates the sample in column i and row j in the temporally previous reference picture.

Next, a filter function that implements the bilinear interpolation described in P.3 is defined. It is assumed that all arguments of the following function are integers and that the bilinear interpolation coefficients, $Ø_x$ and $Ø_y$, are quantized in the range 0, …, P–1 (inclusive).

```
filter(x0, y0, Øx, Øy) {
  return [(P−Øy)·((P−Øx)·prior_sample(x0, y0)+Øx·prior_sample(x0+1, y0))+
          Øy·((P−Øx)·prior_sample(x0, y0+1)+Øx·prior_sample(x0+1, y0+1))+
          P²/2−1+RCRPR]/P²;
}
```

Finally, the method for warping the reference picture to generate a prediction for the current picture can be specified in terms of these functions. The pixels of the prediction picture can be generated in a raster scan order. It is assumed that the values $u_x^L(j)$, $u_y^L(j)$, $u_x^R(j)$, and $u_y^R(j)$ are already calculated and loaded into variables u_x^L, u_y^L, u_x^R, and u_y^R. Defining parameter D as $D = 64H'/P$ and noting that $H' = 2^m$, the sample values of the pixels in the jth line of the resampled field (the topmost line is defined as the 0th line) is obtained by the following pseudo-code:

```
a_x^i = D * P + 2 * (u_x^R − u_x^L);
a_y^i = 2 * (u_y^R − u_y^L);
a_x = u_x^L * S * 2^m + (u_x^R − u_x^L); + D / 2;
a_y = j * D * P + u_y^L * S * 2^m + (u_y^R − u_y^L); + D / 2;
for (i = 0; i < S * H / 2; i++) {
    I_R = a_x /// D;
    J_R = a_y /// D;
    i_R = I_R /// P;
    j_R = J_R /// P;
    Øx = I_R − (i_R * P);
    Øy = J_R − (j_R * P);
    new_ref[i, j] = filter(i_R, j_R, Øx, Øy);
    a_x += a_x^i;
    a_y += a_y^i;
}
```

where all the variables used in this code are integer variables and `new_ref[i,j]` indicates the sample generated for column i and row j in the resampled reference picture. According to the definition of the parameters, all the divisions in this code can be replaced by binary shift operations. For example, when $P = 16$:

```
I_R = a_x /// D;
J_R = a_y /// D;
i_R = I_R /// P;
j_R = J_R /// P;
Øx = I_R − (i_R * P);
Øy = J_R − (j_R * P);
```

can be rewritten, assuming that a_x, a_y, I_R, and J_R are binary-coded integer variables in two's complement representation, as:

```
I_R = a_x >> (m+2);
J_R = a_y >> (m+2);
i_R = I_R >> 4;
j_R = J_R >> 4;
Ø_x = I_R & 15;
Ø_y = J_R & 15;
```

where ">> N_{shift}" denotes a right arithmetic binary shift by N_{shift} bits (N_{shift} is a positive integer), and "&" denotes a bit-wise AND operation.

P.5 Factor-of-4 resampling

Factor-of-4 resampling, which converts both the horizontal and vertical size of the picture by a factor of 2 or $\frac{1}{2}$, is a special case of the resampling algorithm described in P.3. The simplified description of the resampling algorithm for this special case is provided in this clause.

The value of the parameter RCRPR used in Figures P.4 to P.6 is determined by the Rounding Type (RTYPE) bit in MPPTYPE (see 5.1.4.3) as described in P.3. Additionally, "/" in the figures indicates division by truncation.

P.5.1 Factor-of-4 upsampling

The pixel value interpolation method used in factor-of-4 upsampling for internal pixels are shown in Figure P.4. By assuming the existence of pixels outside the picture according to the selected fill mode (see P.2.3 and P.2.4), the same interpolation method is applied for boundary pixels. The interpolation method for boundary pixels when *clip* is selected as the fill mode is shown in Figure P.5. Since precise factor-of-4 upsampling requires x- and y-displacements with at least $\frac{1}{4}$-pixel accuracy, the Warping Displacement Accuracy (WDA) field specified in P.2.1 must be set to "11" or the resampling must be implicitly invoked in order to use this upsampling method.

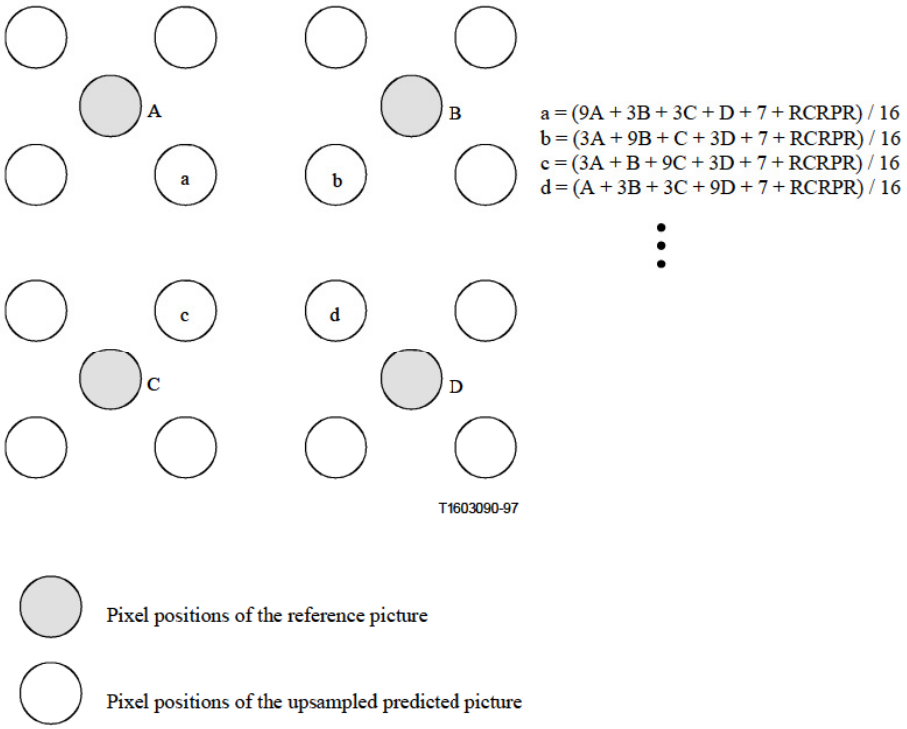


Figure P.4/H.263 – Factor-of-4 upsampling for pixels inside the picture

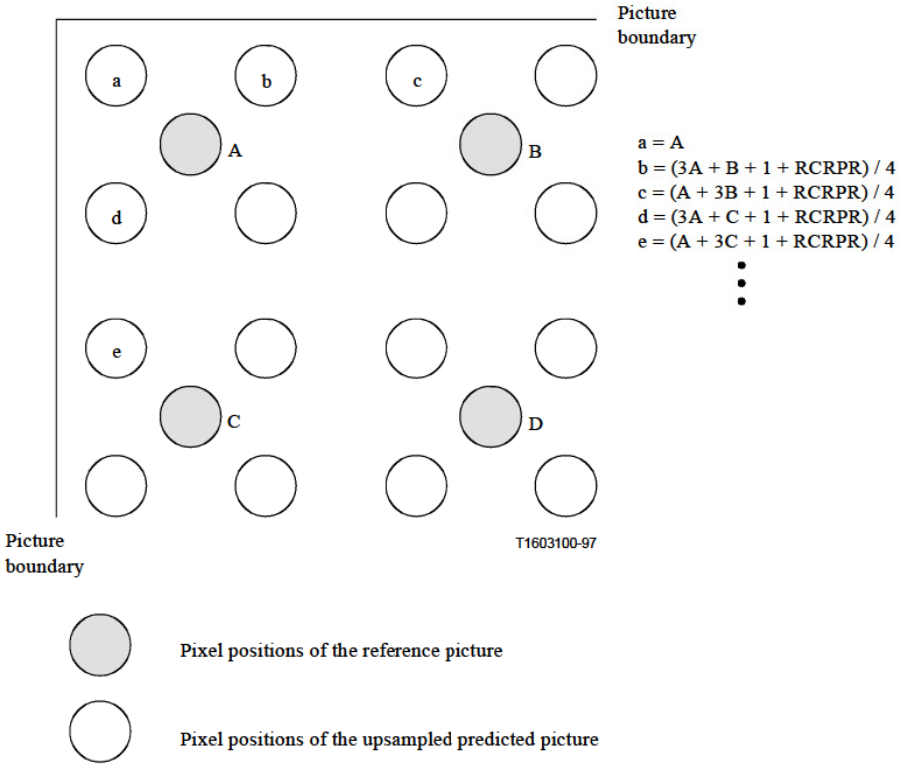


Figure P.5/H.263 – Factor-of-4 upsampling for pixels at the picture boundary (fill mode = *clip*)

P.5.2 Factor-of-4 downsampling

The pixel value interpolation method for factor-of-4 downsampling is shown in Figure P.6. Since x- and y-displacements with $\frac{1}{2}$-pixel accuracy is sufficient for precise factor-of-4 downsampling, both "10" and "11" are allowed as the value of the displacement accuracy (WDA) field (if present) specified in P.2.1.

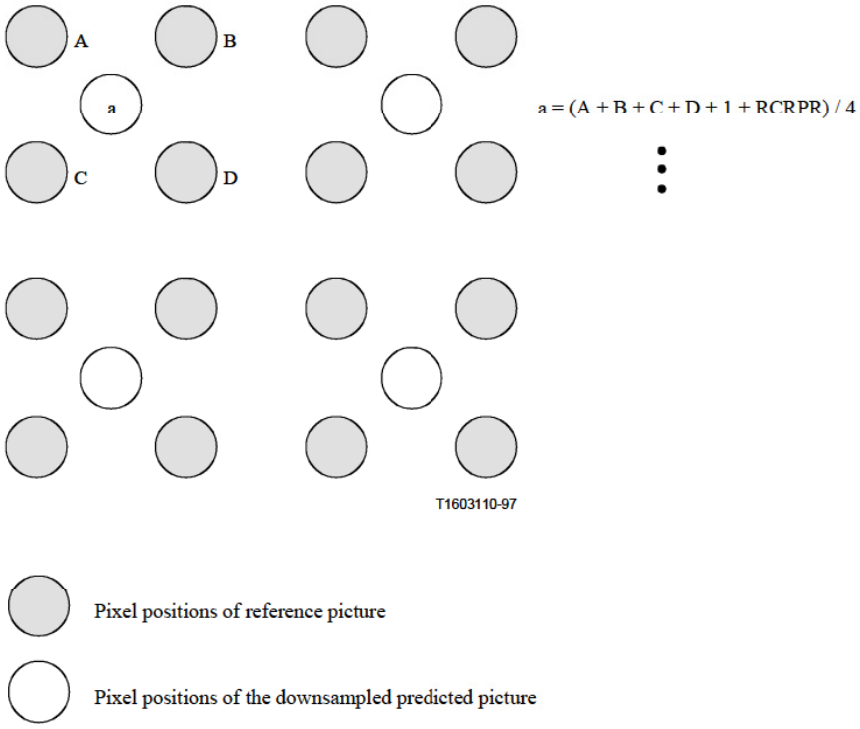


Figure P.6/H.263 – Factor-of-4 downsampling

Annex Q

Reduced-Resolution Update mode

Q.1 Introduction

This annex describes an optional Reduced-Resolution Update mode of this Recommendation. The capability of this mode is signalled by external means (for example, ITU-T Rec. H.245). The use of this mode is indicated in the PLUSPTYPE field of the picture header.

The Reduced-Resolution Update mode is expected to be used when encoding a highly active scene, and provides the opportunity to increase the coding picture rate while maintaining sufficient subjective quality. This mode allows the encoder to send update information for a picture that is encoded at a reduced resolution, while preserving the detail in a higher resolution reference image to create a final image at the higher resolution.

The syntax of the bitstream in this mode is identical to the syntax for coding without the mode, but the semantics, or interpretation of the bitstream is somewhat different. In this mode, the portion of the picture covered by a macroblock is twice as wide and twice as high. Thus, there is approximately one-quarter the number of macroblocks as there would be without this mode. Motion vector data also refers to blocks of twice the normal height and width, or 32 × 32 and 16 × 16 instead of the normal 16 × 16 and 8 × 8. On the other hand, the DCT or texture data should be thought of as describing 8 × 8 blocks on a reduced-resolution version of the picture. To produce the final picture, the texture data is decoded at a reduced resolution and then up-sampled to the full resolution of the picture. After upsampling, the full resolution texture image is added to the (already full resolution) motion compensated image to create the image for display and further reference.

In this mode, a picture which has the horizontal size H and vertical size V as indicated in the picture header is created as a final image for display.

In this mode, a referenced picture used for prediction and created for further decoding has a horizontal size H_R and a vertical size V_R which are the same as in the default mode as defined in 4.1. That is, the H_R and V_R are:

$$H_R = ((H + 15) / 16) * 16$$
$$V_R = ((V + 15) / 16) * 16$$

where H and V are the horizontal size and the vertical size as indicated in the picture header and "/" is defined as division by truncation.

Then in this annex, the texture is coded at a reduced resolution with height and width of H_C and V_C where:

$$H_C = ((H_R + 31) / 32) * 32$$
$$V_C = ((V_R + 31) / 32) * 32$$

and "/" is defined as division by truncation.

If H_C and H_R, or V_C and V_R are not identical to each other, such as in the QCIF format, an extension of the referenced picture is performed, and the picture is decoded in the same manner as if the width and height are H_C and V_C. Then the resulting picture which is tiled by 32 * 32 macroblocks is cropped at the right and the bottom to the width H_R and height V_R, and this cropped picture is stored as a reference picture for further decoding.

If both H and V are the same as those of a resulting picture having the width H_C and the height V_C, this resulting picture is used for display. Otherwise, this resulting picture is further cropped to the size H * V, and the cropped picture here is used for display purpose only.

If the Temporal, SNR, and Spatial Scalability mode (Annex O) or Reference Picture Resampling mode (Annex P) is also used with this option, the source format of the current picture might be different from that of the reference picture. In this case, the resampling of the reference picture shall be performed before decoding.

NOTE – This mode can be used with the Reference Picture Selection mode (see Annex N) without modification, since the reference picture (after possible resampling by the Reference Picture Resampling mode) has the same size as indicated in the current picture header when this option is used.

Q.2 Decoding procedure

Figure Q.1 shows the block diagram of the block decoding in the Reduced-Resolution Update mode.

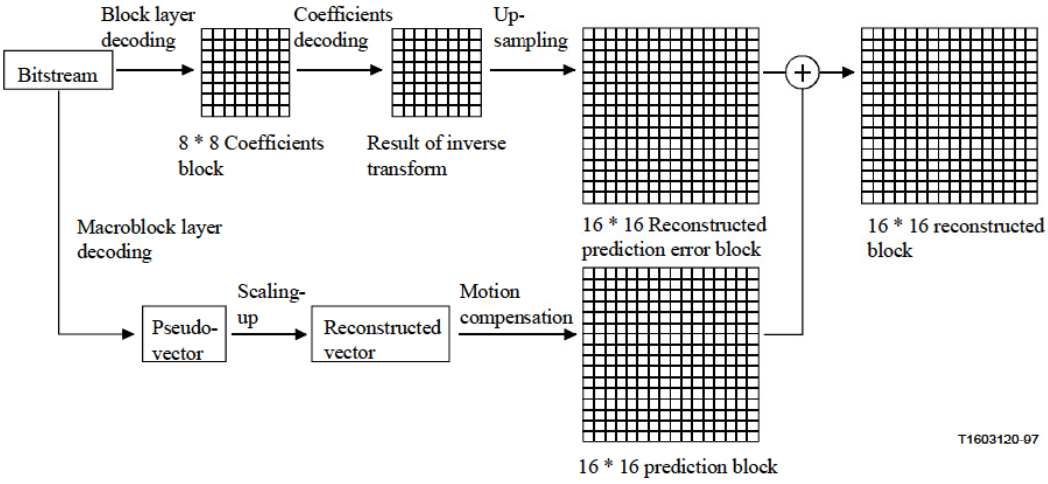


Figure Q.1/H.263 – Block diagram of block decoding in Reduced-Resolution Update mode

The decoding procedure is given in the following subclauses:

Q.2.1 Reference preparation

In some cases, the available reference picture has different size from H_C and V_C. Then, the reference picture shall be converted before decoding procedure according to Q.2.1.1 or Q.2.1.2.

Q.2.1.1 Reference Picture Resampling

If the Temporal, SNR, and Spatial Scalability mode (Annex O) or Reference Picture Resampling mode (Annex P) is also used with this option, the source format of the current picture might be different from that of the reference picture. In this case, the resampling of the reference picture according to each annex shall be performed first.

Q.2.1.2 Extension of Reference Picture

If H_R or V_R is not divisible by 32, such as with the QCIF format, the reference picture is extended. The detailed procedure for this extension is defined in Q.3.

Q.2.2 Macroblock layer decoding

Decoding can be thought of as operating on "enlarged" blocks of size 32 × 32 in luminance and 16 × 16 in chrominance. The texture and motion data for each enlarged block is decoded to create a 32 × 32 motion block and a 32 × 32 texture block, as described in Q.2.2.1 and Q.2.2.2 respectively. These motion and texture blocks are then added as described in Q.2.2.3.

Q.2.2.1 Motion compensation

First, each component of the macroblock motion vector (or four macroblock motion vectors) is formed from MVD (and possibly $MVD_{2\text{-}4}$). If in Improved PB-Frames mode, MV_F and MV_B for B-Picture is also formed from MVDB. The detailed procedure for this motion vector formation is defined in Q.4. If the current picture mode is a B-picture or EP-picture, the motion vector for forward and backward are also obtained according to Q.4.

The motion vector for the two chrominance blocks of the macroblock is obtained from the macroblock motion vector according to 6.1.1. If either the Advanced Prediction mode or the Deblocking Filter mode is in use and thus four motion vectors are defined for the macroblock, the motion vector for both chrominance blocks is obtained from the four motion vectors according to F.2. If in the Improved PB-frames mode, the creation of the chrominance vector is specified in Annex M. If a B-picture or EP-picture is used, the creation of the chrominance vector is specified in Annex O.

Then, a prediction is formed from the motion vector for an INTER macroblock. Four 16 × 16 luminance prediction blocks are obtained from the macroblock motion vector, and two 16 × 16 chrominance prediction blocks are obtained from the chrominance motion vector. For interpolation for subpixel prediction, refer to 6.1.2. If Advanced Prediction mode is also used, an enlarged overlapped motion compensation is performed to obtain four 16 × 16 luminance prediction blocks using enlarged weighting matrices, for which a detailed procedure is defined in Q.5. If the current picture mode is an Improved PB-frame, B-picture, or EP-picture, the prediction is obtained according to the other relevant annexes, except that the size of the predicted blocks is 16 × 16 instead of 8 × 8.

Q.2.2.2 Texture decoding

First, the bitstream of the block layer is decoded according to 5.4. Then, coefficients are decoded and the 8 × 8 reduced-resolution reconstructed prediction error blocks are obtained as the result of inverse transform according to 6.2.

Then the 16 × 16 reconstructed prediction error blocks are obtained by upsampling the 8 × 8 reduced-resolution reconstructed prediction error blocks. For the creation of the edge pixels in each 16 × 16 reconstructed prediction error block, only the pixels which belong to the corresponding block are used. The detailed procedure is defined in Q.6.

Q.2.2.3 Reconstruction of block

For each luminance and chrominance block, a summation of the prediction and prediction error is performed. The procedure is identical to 6.3.1, except that the size of blocks is 16 × 16 instead of 8 × 8. Then the clipping is performed according to the 6.3.2.

Then, a block boundary filter is applied to the boundary pixels of the 16 × 16 reconstructed blocks. The detailed procedure is described in Q.7.

Q.2.3 Picture store

If both H_R and V_R are divisible by 32, such as with the CIF format, the resulting picture reconstructed as described in Q.2.2 is stored as a reference picture as it is for further decoding. Otherwise, such as the QCIF format, the reconstructed picture which is just covered with

32 × 32 macroblocks is cropped at the right and the bottom to the width H_R and height V_R, and this cropped picture is stored as a reference picture for further decoding.

Q.2.4 Display

If both H and V are the same as H_C and V_C, the resulting picture in Q.2.2 is used for display purposes as it is. Otherwise, this resulting picture is further cropped to the size $H \times V$, and the cropped picture is used for display purpose only.

Q.3 Extension of referenced picture

If H_R or V_R is not divisible by 32, such as with the QCIF format, the extension of the referenced picture is performed before decoding macroblock/block layer. The width and the height of the extended reference picture for luminance are the next larger size that would be divisible by 32, and those for chrominance are the next larger size that would be divisible by 16.

NOTE – The width and height of the referenced picture in the default mode are always extended to be divisible by 16 even if the picture format has a width or height that is not divisible by 16, because the picture shall be decoded as if the width or height had the next larger size that would be divisible by 16. See 4.1.

If neither Unrestricted Motion Vector mode, Advanced Prediction mode nor Deblocking Filter mode is used with this option, the extended pixels can be arbitrary values, because the extended pixels will be never used as a reference pixels of the decoded picture to be reconstructed and displayed.

If either Unrestricted Motion Vector mode, Advanced Prediction mode or Deblocking Filter mode is also used with this option, the extension of the referenced picture is performed by duplicating the edge pixel of the referenced picture, in order to ensure the decoding when motion vectors point outside the right and bottom edge of the picture.

For example, if the Reduced-Resolution Update mode is used for a QCIF, the width of the referenced picture is 176 and the height is 144, which are not divisible by 32. In order to cover a QCIF picture with 32 × 32-sized macroblocks, the number of macroblock row should be 6, and the number of macroblock column should be 5. Therefore, the width of the extended referenced picture is 192 and the height is 160.

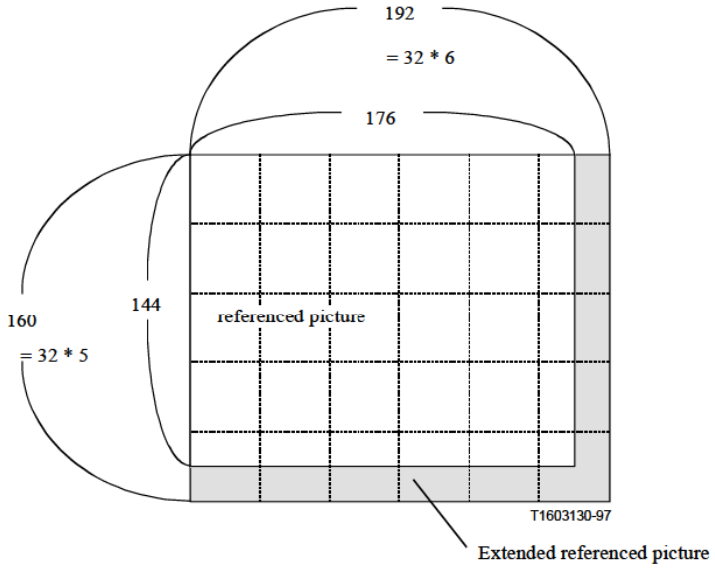


Figure Q.2/H.263 – Extension of referenced picture for QCIF picture size

The extension of the referenced picture in QCIF is illustrated in Figure Q.2. The extended referenced picture for luminance is given by the following formula:

$$R_{RRU}(x,\ y) = R(x',\ y')$$

where:

x, y = spatial coordinates of the extended referenced picture in the pixel domain

x', y = spatial coordinates of the referenced picture in the pixel domain

$R_{RRU}(x, y)$ = pixel value of the extended referenced picture at (x, y)

$R(x', y')$ = pixel value of the referenced picture at (x', y')

$$x' = \begin{cases} 175 & \text{if } x > 175 \text{ and } x < 192 \\ x & \text{otherwise} \end{cases}$$

$$y' = \begin{cases} 143 & \text{if } y > 143 \text{ and } y < 160 \\ y & \text{otherwise} \end{cases}$$

The referenced pictures for chrominance is also extended in the same manner.

Q.4 Reconstruction of motion vectors

In Reduced-Resolution Update mode, the motion vector range is enlarged to approximately double size in both the horizontal and vertical directions. In order to realize an enlarged range using the VLC for MVD defined in Table 14, each vector component is restricted to be only half-pel or zero value. Therefore the range of each motion vector component is [–31.5, 30.5] in the default Reduced-Resolution Update mode. If the Unrestricted Motion Vector mode is used, the vector range [–limit, limit] as defined in D.2 applies to the pseudo-motion vectors and translates into [–(2 * limit–0.5), 2 * limit–1.5] for the motion vectors. For CIF this means that the pseudo-motion vector range is [–32, 31.5] and the motion vector range is [–63.5, 62.5]. If the UUI field is set to "01", the motion vectors are unlimited. However, the motion vectors (not just the pseudo-motion vectors) are always limited to point not more than 15 pixels outside the coded area as described in D.1.1. Figure Q.3 illustrates the possible positions of macroblock motion vector or four motion vector predictions around the (0, 0) vector value. The dashed lines indicate the integer coordinates.

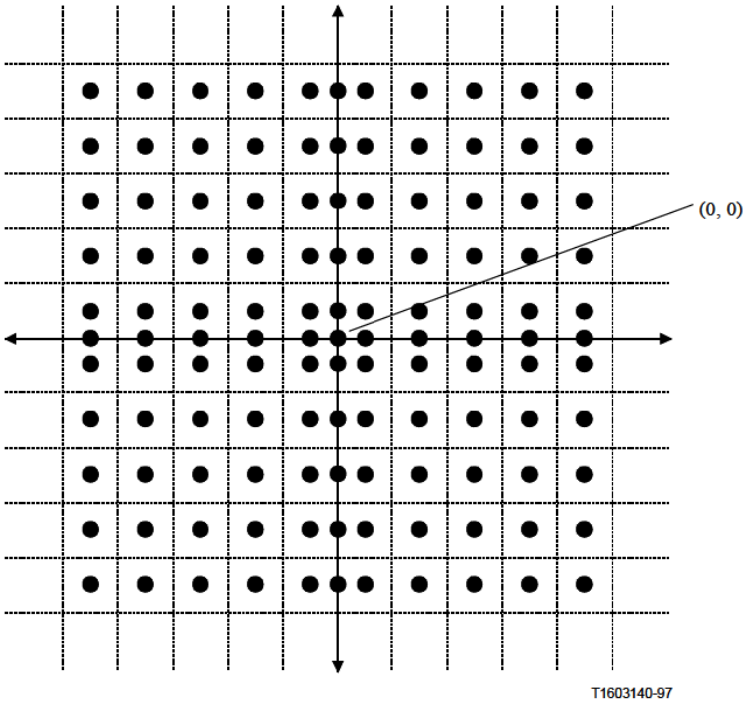


Figure Q.3/H.263 – Reconstruction of motion vector

For the macroblock using differential motion vectors in a B-picture, the motion vector for forward and backward prediction are obtained independently. In the Reduced-Resolution Update mode, the motion vector component MV_C for a luminance block is reconstructed from MVD and $MVD_{2\text{-}4}$ as follows:

1) Pseudo-prediction vector component **pseudo-P_C** is created from the prediction vector component **P_C**.

   pseudo-$P_C = 0$ if $P_C = 0$

   pseudo-P_C = sign(P_C) * (|P_C| + 0.5) / 2.0 if $P_C \neq 0$

   "/" indicates floating-point division (without loss of accuracy). The prediction vector component P_C is defined as the median value of the vector components MV1, MV2 and MV3 as defined in 6.1.1 and F.2.

2) Pseudo-macroblock vector component **pseudo-MV_C** is obtained by adding the motion vector differences MVD (and $MVD_{2\text{-}4}$) from Table 14 to the pseudo-P_C.

   In the default Reduced-Resolution Update mode, the value of **pseudo-MV_C** is restricted to the range [–16, 15.5]. Only one of the pair will yield a pseudo-MV_C falling within the permitted range. The procedure is performed in a similar way as defined in 6.1.1.

   If the Unrestricted Motion Vector mode is also used with the Reduced-Resolution Update mode, **pseudo-MV_C** is obtained by adding the motion vector differences MVD (and $MVD_{2\text{-}4}$) from Table D.3.

   If four motion vectors are present, the procedure is performed in a similar way as defined in F.2.

3) Motion vector component $\mathbf{MV_C}$ is obtained from pseudo-MV_C in the following formula:

$MV_C = 0$ if pseudo-$MV_C = 0$

$MV_C = \text{sign}(\text{pseudo-}MV_C) * (2.0 * |\text{pseudo-}MV_C| - 0.5)$ if pseudo-$MV_C \neq 0$

As a result, each vector component is restricted to have a half-integer or zero value, and the range of each motion vector component is enlarged to approximately twice the pseudo-motion vector range.

4) If the current picture mode is an Improved PB-frame, or when the MBTYPE indicates the direct mode in a B-picture, the motion vector components $\mathbf{MV_F}$ and/or $\mathbf{MV_B}$ for forward and backward prediction are created.

First, pseudo-motion vector components **pseudo-$\mathbf{MV_F}$** and/or **pseudo-$\mathbf{MV_B}$** are calculated based on the rules for the prediction modes defined in Annexes O or M.

In the case of the Bidirectional prediction in the Improved PB-frames mode (see M.2.1) or when the MBTYPE indicates the direct mode in a B-picture (see O.5.2), pseudo-MV_F and pseudo-MV_B are calculated from **pseudo-$\mathbf{MV_D}$** and **pseudo-$\mathbf{MV_C}$** assuming that pseudo-MV_D is zero and pseudo-MV_C is MV, as defined in Annexes G and M.

In the case of forward prediction in the Improved PB-frames mode (see M.2.2), **pseudo-$\mathbf{MV_{DB}}$** is obtained by decoding the variable length code MVDB according to Table 12. Then, pseudo-MV_F is obtained by adding the pseudo-MV_{DB} to the **pseudo-Predictor**. In order to form the pseudo-Predictor, the predictor obtained according to the procedure defined in M.2.2 is converted to the pseudo-Predictor vector according to the formula defined in item 1 of this clause.

In the case of backward prediction in the Improved PB-frames mode (see M.2.3), pseudo-MV_B is set to zero.

Then, the motion vectors MV_F and/or MV_B for forward and backward prediction are obtained from pseudo-MV_F and/or pseudo-MV_B according to the formula defined in item 3 of this clause.

Q.5 Enlarged overlapped motion compensation for luminance

If Advanced Prediction mode is also used with Reduced-Resolution Update mode, enlarged matrices of weighting values are used to perform the overlapped motion compensation. Except that the size of each block and weighting matrices is 16×16, the procedure of the creation of each prediction block is identical to the description in F.3.

The enlarged matrices of weighting values for the 16 × 16 luminance prediction are given in Figures Q.4, Q.5 and Q.6.

4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	6	6	6	6	6	6	6	6	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4
4	4	5	5	5	5	5	5	5	5	5	5	5	5	4	4

Figure Q.4/H.263 – Weighting values, H_0, for prediction with motion vector of current 16 × 16 luminance block

2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
1	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2

Figure Q.5/H.263 – Weighting values, H_1, for prediction with motion vector of current 16 × 16 luminance blocks on top or bottom of current 16 × 16 luminance block

2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	2	2	1	1	1	1	1	1	1	1	2	2	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	2	2

Figure Q.6/H.263 – Weighting values, H_2, for prediction with motion vector of 16 × 16 luminance blocks to the left or right of current 16 × 16 luminance block

Q.6 Upsampling of the reduced-resolution reconstructed prediction error

The 16 × 16 reconstructed prediction error block is obtained by upsampling the 8 × 8 reduced-resolution reconstructed prediction error block. In order to realize a simple implementation, filtering is closed within a block which enables to perform an individual upsampling on block basis. Figure Q.7 shows the positioning of samples. The upsampling procedure for the luminance and chrominance pixels which are inside the 16 × 16 reconstructed prediction error blocks is defined in Q.6.1. For the creation of the luminance and chrominance pixels which are at the boundary of 16 × 16 reconstructed prediction error block, the procedure is defined in Q.6.2. Chrominance blocks as well as luminance blocks are up-sampled. The symbol "/" in Figures Q.8 and Q.9 indicates division by truncation.

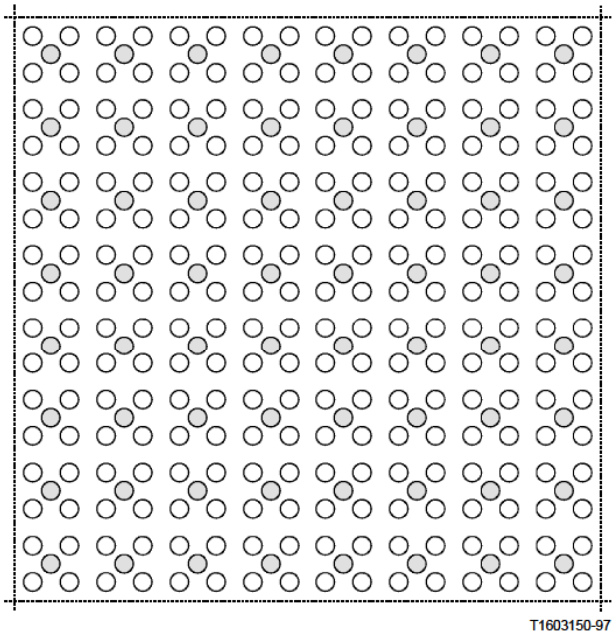


Figure Q.7/H.263 – Positioning of samples in 8 × 8 reduced-resolution reconstructed prediction error block and 16 × 16 reconstructed prediction error block

Q.6.1 Upsampling procedure for the pixels inside a 16 × 16 reconstructed prediction error block

The creation of reconstructed prediction error for pixels inside block is described in Figure Q.8. "/" indicates division by truncation.

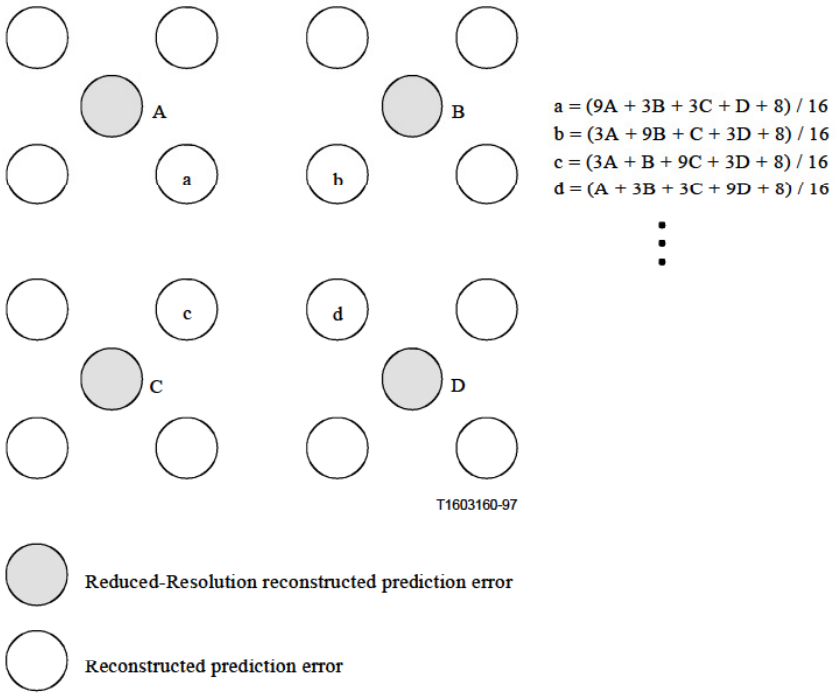


Figure Q.8/H.263 – Creation of reconstructed prediction error for pixels inside block

Q.6.2 Upsampling procdedure for the pixels at the boundary of 16 × 16 reconstructed prediction error block

The creation of reconstructed prediction error for pixels of a 16 × 16 block is shown in Figure Q.9.

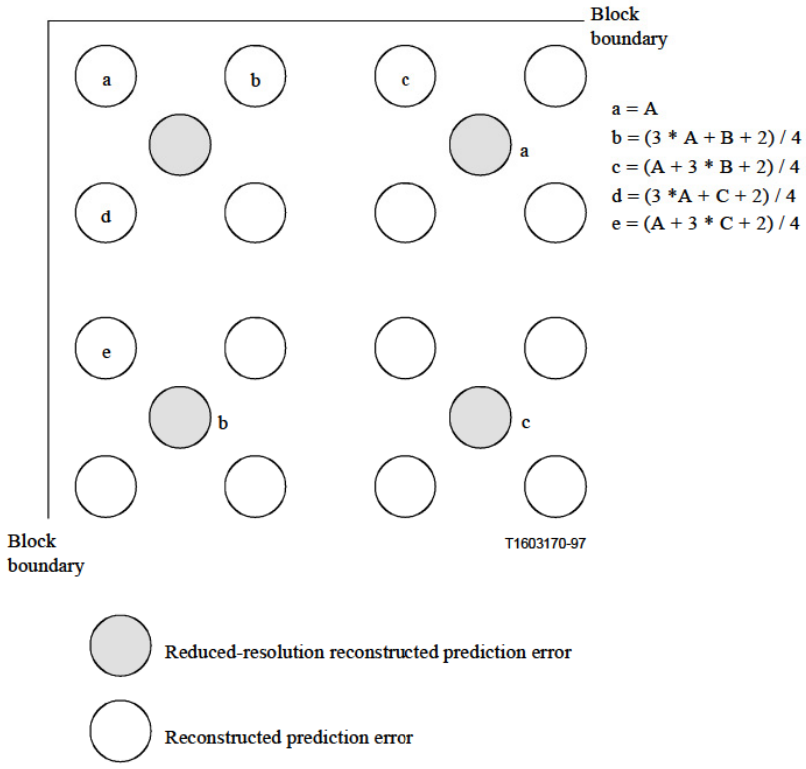


Figure Q.9/H.263 – Creation of reconstructed prediction error for pixels at the block boundary

Q.7 Block boundary filter

The filter operations are performed along the edges of the 16 × 16 reconstructed blocks at the encoder as well as on the decoder side. There are two alternative of filtering, depending on whether Deblocking Filter mode is used or not.

The default filtering in Reduced-Resolution Update mode is performed according to Q.7.1.

If Deblocking Filter mode is also used with Reduced-Resolution Update mode, the filtering is performed according to Q.7.2.

In both cases, filtering is performed on the complete reconstructed image data before storing the data in the picture store for future prediction. No filtering is performed across picture edges, slice edges in Slice Structured mode (see Annex K), or GOB boundaries having GOB headers present in Independent Segment Decoding mode (see Annex R). Chrominance as well as luminance data is filtered.

Q.7.1 Definition of the default block boundary filter

In the Reduced-Resolution Update mode, the default filtering is performed according to this subclause.

If A and B are two pixel values on a line – horizontal or vertical – of the reconstructed picture, and A belongs to one 16 × 16 block called block1 whereas B belongs to a neighboring 16 × 16 block called block2 which is to the right or below of block1. Figure Q.10 shows examples for the position of these pixels.

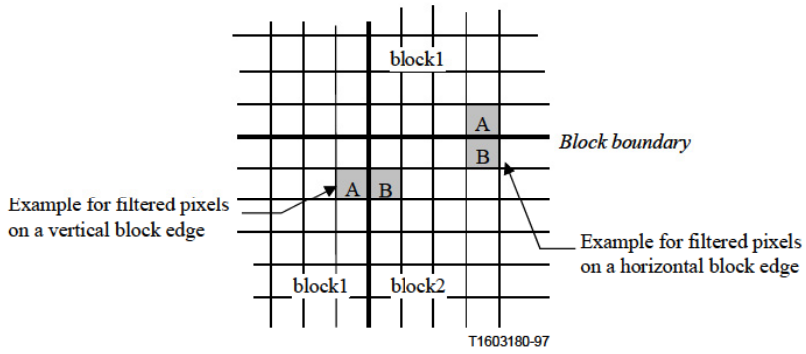


Figure Q.10/H.263 – Default block boundary filter

One of the following conditions must be fulfilled in order to turn the filter on for a particular edge:

- block1 belongs to a coded macroblock (COD==0 || MB-type == INTRA); **or**
- block2 belongs to a coded macroblock (COD==0 || MB-type == INTRA).

A shall be replaced by A1 and B shall be replaced by B1. "/" indicates division by truncation.

$$A1 = (3 * A + B + 2) / 4$$

$$B1 = (A + 3 * B + 2) / 4$$

The order of edges where filtering is performed is identical to the description provided in J.3.

Q.7.2 Definition of the block boundary filter when Deblocking Filter mode is used

If the Deblocking Filter mode (see Annex J) is used with the Reduced-Resolution Update mode, the filtering which is defined in Annex J with one modification is performed on the boundary pixels of 16 × 16 luminance and chrominance blocks, in place of the filtering described in Q.7.1. The one modification of the filtering in Annex J is that the parameter STRENGTH is given the value of positive infinity. This implies that the function UpDownRamp(x, STRENGTH) defined in J.3 becomes a linear function of x.

As a result, the procedure of the deblocking filter described in J.3 is redefined in the following manner:

B1 = clip(B + d1)

C1 = clip(C – d1)

A1 = A – d2

D1 = D + d2

d1 = (A–4B+4C–D) / 8

d2 = clipd1((A – D) / 4, d1/2)

Annex R

Independent Segment Decoding mode

R.1 Introduction

This annex describes the optional Independent Segment Decoding mode of this Recommendation, which allows a picture to be decoded without the presence of any data dependencies across slice boundaries or across GOB boundaries having non-empty GOB headers. The use of this mode is indicated in the PLUSPTYPE field of the picture header. The capability to use this optional mode is negotiated by external means (for example, ITU-T Rec. H.245).

When the use of this mode is indicated, the video picture segment boundaries (as defined by the boundaries of the slices or the upper boundaries of the GOBs for which GOB headers are sent, or the boundaries of the picture, whichever bounds a region in the smallest way) are treated as picture boundaries when decoding, including the treatment of motion vectors which cross those boundaries (which result in boundary extrapolation when the Unrestricted Motion Vector mode, the Advanced Prediction mode, the Deblocking Filter mode, or the Temporal, SNR, and Spatial Scalability mode are in use, and which are prohibited when none of those optional modes are in use).

R.2 Mode operation

A video picture segment is defined by the following.

If the Slice Structured mode (see Annex K) is not in use, then one GOB or a plural number of consecutive GOBs forms one video picture segment. The location of the top of each video picture segment is indicated by the presence of a non-empty GOB header for which the border of the video picture segment lies just above the macroblocks in the GOB for which a header is present, or the top of the picture, whichever is lower. The location of the bottom of each video picture segment is defined by the top of the next video picture segment, or the bottom of the picture, whichever is uppermost.

If the Slice Structured mode (see Annex K) is in use, then each slice forms one video picture segment.

In the Independent Segment Decoding mode, each video picture segment is decoded with complete independence from all other video picture segments, and is also independent of all data outside the same video picture segment location in the reference picture(s). This includes:

1) no use of motion vectors outside of the current video picture segment for motion vector prediction (as in 6.1.1);
2) no use of motion vectors outside of the current video picture segment as remote motion vectors for overlapped block motion compensation when the Advanced Prediction mode is in use (see F.3);
3) no deblocking filter operation across video picture segment boundaries (see J.3);
4) no use of motion vectors which reference data outside the current video picture segment unless the Unrestricted Motion Vector mode (see Annex D), the Advanced Prediction mode (see Annex F), the Deblocking Filter mode (see Annex J) or the Temporal, SNR, and Spatial Scalability mode (see Annex O) are in use; in which case the borders of the current video picture segment in the prior picture are extrapolated as described in Annex D to form predictions of the pixels which reference the out-of-bounds region;
5) no bilinear interpolation across the boundaries of the ¼-size or ½-size region corresponding to the current video picture segment for upward prediction in spatial scalability EI- and EP-pictures (as defined in Annex O);

6) when the Reduced-Resolution Update mode (see Annex Q) is in use, no block boundary filter operation across video picture segment boundaries;

7) no use of the Reference Picture Resampling mode with the Independent Segment Decoding mode.

R.3 Constraints on usage

Certain restrictions are placed on the use of other aspects of the video coding syntax when the Independent Segment Decoding mode is in use. These restrictions are made to prevent two pathological cases which otherwise would make operation of the Independent Segment Decoding mode difficult.

R.3.1 Constraint on segment shapes

In the use of the Slice Structured mode (Annex K) without the use of the Rectangular Slice submode (see K.1), there can arise cases in which the shape of a video picture segment may be non-convex (having "inside corners", or even comprising two distinct and separated regions of the picture).

Therefore, the Independent Segment Decoding mode shall not be used with the Slice Structured mode without the simultaneous use of the Rectangular Slice submode of the Slice Structured mode (see Annex K). This constraint is mandated to prevent the need for difficult special-case treatment in order to determine how and when to perform extrapolation of each video picture segment.

R.3.2 Constraint on changes of segment shapes

If the shape of the video picture segments were allowed to change in any way from picture to picture in the bitstream, there could arise cases in which the bitstream would be difficult to decode. This is because in such cases the bitstream content is not sufficient to determine the shape of each video picture segment prior to the possible appearance of motion vectors in the bitstream which require knowledge of the video picture segment shape for proper interpretation.

Therefore, when the Independent Segment Decoding mode is in use, the video picture segmentation for all pictures and frames allowing temporal prediction (i.e., all P-, B-, and EP-pictures and all Improved PB-frames) shall be the same as that used in its temporal reference picture. Also, when the Independent Segment Decoding mode is in use, the video picture segmentation for all EI-pictures shall either be the same or shall differ only by sub-dividing the video picture segmentation used in its reference picture. Also, the Independent Segment Decoding mode shall not be used in any picture or frame which uses reference pictures (all picture types except INTRA) unless the Independent Segment Decoding mode is also used in all of the reference picture(s) for the current picture. As a result of this constraint, the shape of the video picture segments in the Independent Segment Decoding mode shall never change from picture to picture except as changed in I- and EI-pictures (and the manner in which EI-pictures can change segmentation is itself also somewhat constrained).

Annex S

Alternative INTER VLC mode

S.1 Introduction

This annex describes an optional Alternative INTER VLC mode of this Recommendation, which improves the efficiency of inter-picture coding when significant changes are evident in the picture. This efficiency improvement is obtained by allowing some VLC codes originally designed for INTRA pictures to be used for some INTER picture coefficients and CBPY data as well. The use of this mode is indicated in the PLUSPTYPE field of the picture header. The capability to use this optional mode is negotiated by external means (for example, ITU-T Rec. H.245). The mode contains two syntax alterations, one for the encoding of INTER coefficients and another for the encoding of the INTER CBPY values.

S.2 Alternative INTER VLC for coefficients

The concept behind the design of the INTRA VLC table of Annex I is to use the same codewords as in the original INTER VLC but with a different interpretation of LEVEL and RUN. The INTRA VLC is better suited in cases where there are many and/or large-valued coefficients.

The INTRA VLC is constructed so that codewords have the same value for LAST (0 or 1) in both the INTER and INTRA tables. The INTRA table is therefore produced by "reshuffling" the meaning of the codewords with the same value of LAST. Furthermore, for events with large |LEVEL| the INTRA table uses a codeword which in the INTER table has a large RUN. In INTER blocks having a large number of large-magnitude coefficients, it can sometimes be more efficient to use the INTRA table than the INTER table, and in some such cases the choice of the VLC table can be apparent to the decoder since decoding using the INTER table would result in RUN values so large as to indicate the presence of more than 64 coefficients for a block. Under these circumstances, the INTRA table can be used to improve the efficiency of INTER coding.

S.2.1 Encoder action

The encoder may use the INTRA VLC table for coding an INTER block whenever the decoder can detect its use – in other words, whenever decoding using the INTER VLC table would cause coefficients outside the 64 coefficients of a block to be addressed.

The encoder would normally choose to use the INTRA VLC table for coding an INTER block only when the above condition is satisfied and also when the INTRA VLC usage results in fewer bits than the INTER VLC for the same coefficient values. This will often be the case when there are many large coefficients, due to the way the INTRA VLC was produced (since the ordinary INTER VLC table contains long run-lengths for the same codewords in which the INTRA VLC table contains large coefficient amplitudes).

S.2.2 Decoder action

The decoding process is as follows:

1) The decoder first receives all coefficient codes of a block.
2) The codewords are then interpreted assuming that INTER VLC is used. If the addressing of coefficients stays inside the 64 coefficients of a block, the VLC decoding is finished.
3) If coefficients outside the block are addressed, the codewords shall be interpreted according to the INTRA VLC.

S.3 Alternative INTER VLC for CBPY

The INTER CBPY codewords (Table 12) are designed with the assumption that there are more Y blocks with all zero coefficients than there are with at least one non-zero coefficient. When both C_B and C_R blocks have at least one non-zero coefficient, i.e., $CBPC_5 = CBPC_6 = 1$, this assumption no longer holds. For this reason, when the Alternative INTER VLC mode is in use, the CBPY codewords as defined in Table 12 for INTRA macroblocks shall also be used for INTER macroblocks whenever $CBPC_5 = CBPC_6 = 1$.

Annex T

Modified Quantization mode

T.1 Introduction

This annex describes an optional Modified Quantization mode of this Recommendation, which modifies quantizer operation. The use of this mode is indicated in the PLUSPTYPE field of the picture header. The capability to use this optional mode is negotiated by external means (for example, ITU-T Rec. H.245).

This mode includes four key features:

1) The bit-rate control ability for encoding is improved by altering the syntax for the DQUANT field.
2) Chrominance fidelity is improved by specifying a smaller step size for chrominance than that for luminance data.
3) The range of representable coefficient values is extended to allow the representation of any possible true coefficient value to within the accuracy allowed by the quantization step size.
4) The range of quantized coefficient levels is restricted to those which can reasonably occur, to improve the detectability of errors and minimize decoding complexity.

T.2 Modified DQUANT Update

This mode modifies the semantics of the DQUANT field. With this mode, it is possible to use DQUANT either to modify QUANT by plus or minus a small amount, or to signal any specific new value for QUANT. The size of the small amount of modification depends on the current value of QUANT. By use of this mode, more flexible control of the quantizer step size can be specified in the DQUANT field.

The codeword for DQUANT in this mode is no longer a two-bit fixed length field. It is a variable length field which can either be two bits or six bits in length. Whether it is two or six bits depends on the first bit of the code. The description below is therefore split into two sections, depending on the first bit.

T.2.1 Small-step QUANT alteration

When the first bit of the DQUANT field is 1, only one additional bit is sent in DQUANT. The single additional bit is used to modify QUANT by a differential value. The change in the value of QUANT is dependent on the second bit of DQUANT and on the prior value of QUANT, as shown in Table T.1.

Example: If the previous value of QUANT is 29 and DQUANT is signalled with the codeword "11", then the differential value is +2, and thus the resulting new QUANT value is 31.

Table T.1/H.263 – Semantics of small-step QUANT alteration

	Change of QUANT	
Prior QUANT	**DQUANT = 10**	**DQUANT = 11**
1	+2	+1
2-10	–1	+1
11-20	–2	+2
21-28	–3	+3
29	–3	+2
30	–3	+1
31	–3	–5

T.2.2 Arbitrary QUANT selection

When the first bit of the DQUANT field is 0, five additional bits are sent in DQUANT. The following five bits represent a new QUANT as defined in 5.1.19.

Example: Regardless of the current value of QUANT, if DQUANT is signalled with the code word '001111', then the new value of QUANT is 15.

T.3 Altered quantization step size for chrominance coefficients

When the modified quantization mode is in use, the quantization parameter of the chrominance coefficients is different from the quantization parameter of the luminance. The luminance quantization parameter is signalled in the bitstream. It is called QUANT. When this mode is in use, a different quantization parameter termed QUANT_C is used for the inverse quantization of chrominance coefficients. The relation between QUANT and QUANT_C is given in Table T.2. If the Deblocking Filter mode (see Annex J) is in use, QUANT_C shall also be used for the application of the deblocking filter to the chrominance data. Whenever QUANT is discussed herein in any other context, it shall mean the quantization step size of the luminance.

Table T.2/H.263 – Relationship between QUANT and QUANT_C

Range of QUANT	**Value of QUANT_C**
1-6	QUANT_C = QUANT
7-9	QUANT_C = QUANT – 1
10-11	9
12-13	10
14-15	11
16-18	12
19-21	13
22-26	14
27-31	15

T.4 Modified coefficient range

When the Modified Quantization mode is in use, quantized DCT coefficients having quantization level magnitudes greater than 127 can be represented in the bitstream. This has two advantages:

1) Encoder performance is improved by allowing the true full range of possible coefficient values to be represented.

2) Encoder complexity is reduced by eliminating the need to increase the quantization step size upon encountering certain large coefficient values which would otherwise be unrepresentable.

It is possible that the correct value of a DCT coefficient prior to quantization in the encoder will have a magnitude as high as 2040. Thus, a range of –127 to +127 for LEVEL is insufficient to cover the entire range of possible coefficient values whenever the quantization parameter QUANT or QUANT_C is less than 8. The expanded coefficient range broadens the range of LEVEL to allow any true coefficient value to be more properly encoded.

When the Modified Quantization mode is in use, the meaning of the LEVEL field following an ESCAPE code (0000 011, as per 5.4.2) is altered. In this mode, rather than being forbidden, the bit sequence 1000 0000 is used to represent an EXTENDED-ESCAPE code. An AC coefficient of magnitude greater than 127 is represented by sending an EXTENDED-ESCAPE code, immediately followed by a fixed-length EXTENDED-LEVEL field of 11 bits. An extended coefficient value is encoded into the EXTENDED-LEVEL field by taking the least significant 11 bits of the two's-complement binary representation of LEVEL and cyclically rotating them to the right by 5-bit positions. This rotation is necessary to prevent start code emulation. The cyclic rotation is illustrated in Figure T.1.

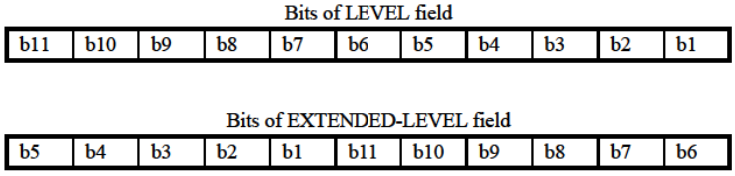


Figure T.1/H.263 – Cyclic rotation of coefficient representation

T.5 Usage restrictions

When the Modified Quantization mode is in use, certain restrictions are placed on the encoded coefficient values. This has several benefits:

1) The detectability of bit errors is improved by prohibiting certain unreasonable coefficient values, thus allowing these values to be recognized as bit errors by the decoder; and

2) Decoder complexity is reduced by reducing the wordlength necessary for inverse quantization prior to clipping;

3) Start-code emulation is prevented for coefficients encoded using the EXTENDED-ESCAPE mechanism described in T.4.

These restrictions are as follows. When the Modified Quantization mode is in use:

1) For any coefficient, the reconstruction level magnitude |REC| produced by the inverse quantization process described in 6.2.1 using the current value of QUANT or QUANT_C as appropriate, and the encoded value of LEVEL, shall be less than 4096. This additional restriction applies to all coefficients, regardless of whether the coefficient is sent using the EXTENDED-ESCAPE mechanism or not.

2) The bitstream shall not use either the normal ESCAPE code or the EXTENDED-ESCAPE code to encode a combination of LAST, RUN, and LEVEL for which there exists a codeword entry in the applicable VLC table, which is either Table 16 (see 5.4.2) or Table I.2 (see I.3).

3) The EXTENDED-ESCAPE code shall be used only when the quantization parameter for the coefficient (QUANT or QUANT_C) is less than eight (8).

4) The EXTENDED-ESCAPE code shall be used only when it is followed by an EXTENDED-LEVEL field representing a value of LEVEL which is outside of the range –127 to +127.

Annex U

Enhanced reference picture selection mode

U.1 Introduction

This annex describes the optional Enhanced Reference Picture Selection (ERPS) mode of this Recommendation. The capability to use this optional mode is negotiated by external means (for example, ITU-T Rec. H.245). The amount of picture memory accommodated in the decoder for ERPS operation should also be signalled by external means. The use of this mode shall be indicated by setting the formerly reserved bit 16 of the optional part of the PLUSPTYPE (OPPTYPE) to "1". The mode provides benefits for both error resilience and coding efficiency by using a memory buffer of reference pictures.

A sub-mode of the ERPS mode is specified for Sub-Picture Removal. The purpose of Sub-Picture Removal is to reduce the amount of memory required to store multiple reference pictures. The memory reduction is accomplished by specifying the partitioning of each reference picture into smaller rectangular units called sub-pictures. The encoder can then indicate to the decoder that specific sub-picture areas of specific reference pictures will not be used as a reference for the prediction of subsequent pictures, thus allowing the memory allocated in the decoder for storing these areas to be used to store data from other reference pictures. The support for this sub-mode and the allowed fragmentation of the picture memory into minimum picture units (MPUs) for Sub-Picture Removal as defined herein is also negotiated by external means (for example, ITU-T Rec. H.245).

A sub-mode of the ERPS mode is specified for enabling two-picture backward prediction in B pictures. This sub-mode can enhance performance by providing encoders for B pictures not only with an ability to use multiple references for forward prediction, but also to use more than one reference picture for backward prediction. The support for this sub-mode is negotiated by external means (for example, ITU-T Rec. H.245).

For error resilience, the ERPS mode can use backward channel messages, which are signalled by external means (for example, ITU-T Rec. H.245) sent from a decoder to an encoder to inform the encoder which pictures or parts of pictures have been incorrectly decoded. The ERPS mode provides enhanced performance compared to the Reference Picture Selection (RPS) mode defined in Annex N. It shall not be used simultaneously with the RPS mode. (It can be used in such a way as to provide essentially the same functionality as the RPS mode.)

For coding efficiency, motion compensation can be extended to prediction from multiple pictures. The extension of motion compensation to multi-picture prediction is achieved by extending each motion vector by a picture reference parameter that is used to address a macroblock or block prediction region for motion compensation in any of the multiple reference pictures. The picture

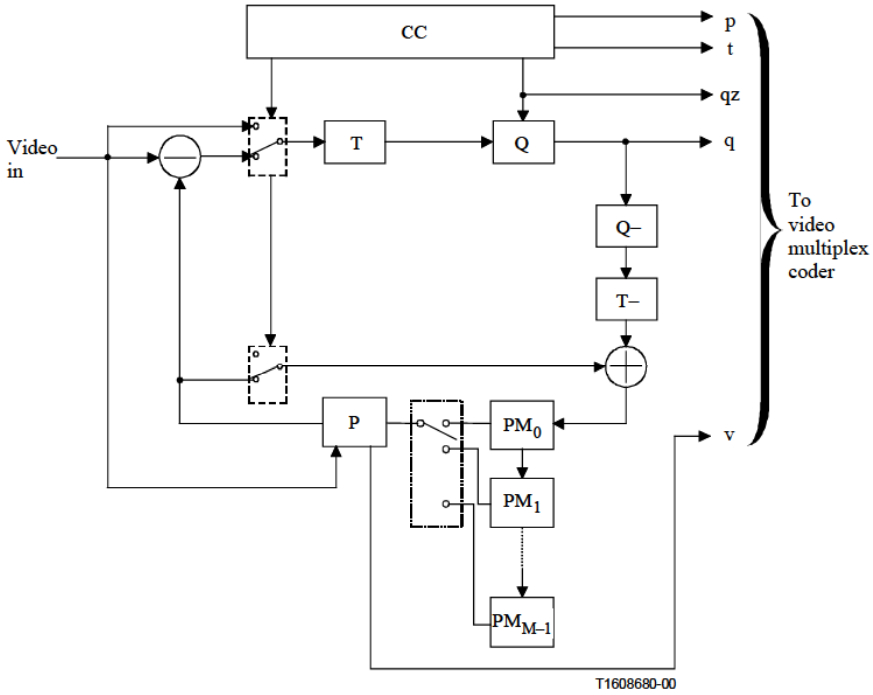
reference parameter is a variable length code specifying a relative buffer index. The reference pictures are assembled in a buffering scheme that is controlled by the encoder.

The ERPS mode shall not be used with the Syntax-based Arithmetic Coding mode (see Annex E) or the Data Partitioned Slice mode (see Annex V).

Once activated, the ERPS mode shall not be inactivated in subsequent pictures in the bitstream unless the initial inactivation occurs in an I or EI picture and any reactivation is also in an I or EI picture and is accompanied by a buffer reset (RESET equal to "1"). If inactivated, the entire contents of the ERPS multi-picture buffer shall be set to "unused" status.

U.2 Video source coding algorithm

The source coder of this mode is shown in generalized form in Figure U.1. This figure shows a structure that uses a number of picture memories.



T	Transform
Q	Quantizer
P	Picture Memory with motion compensated variable delay
PM	Picture Memory
CC	Coding control
p	Flag for INTRA/INTER
t	Flag for transmitted or not
qz	Quantizer indication
q	Quantizing index for transform coefficients
v	Motion vector

Figure U.1/H.263 – Source coder for Enhanced Reference Picture Selection mode

The video source coding algorithm can be extended to multi-picture motion compensation. Enhanced coding efficiency may be achieved by allowing reference picture selection on the macroblock level. A picture buffering scheme with relative indexing is employed for efficient addressing of pictures in the multi-picture buffer. The multi-picture buffer control may work in two distinct types of operation.

In the first of these two types of operation, a "Sliding Window" over time can be accommodated by the buffer control unit. In such a buffering scheme using M picture memories $PM_0...PM_{M-1}$, the most recent preceding (up to M) decoded and reconstructed pictures are stored in the picture memories and can be used as references for decoding. If the number of pictures maximally accommodated by the multi-picture buffer corresponds to M, the motion estimation when coding a picture m, if $0 \le m \le M-1$, can utilize m pictures. When coding a picture $m \ge M$, the maximum number of pictures M can be used. Alternatively, a second "Adaptive Memory Control" type of operation can be used for a more flexible and specific control of the picture memories than with the simple "Sliding Window" scheme.

The operation of the ERPS mode results in the assignment of "unused" status to some pictures or sub-picture areas of pictures that have been sent to the decoder. Once some picture or area of a picture has been assigned to "unused" status, the bitstream shall not contain any data that causes a reference to any "unused" area for the prediction of subsequent pictures. By managing the assignment of "unused" status to previous pictures, the encoder shall ensure that sufficient memory is available in the decoder to store all data needed for the representation of subsequent pictures. The overall buffer size and structure is conveyed to the decoder in the bitstream, and the encoder shall control the buffer such that the specified total capacity is not exceeded by stored picture data that has not been assigned to "unused" status.

The source coder may select one or several of the picture memories to suppress temporal error propagation caused by inter-picture coding. The Independent Segment Decoding mode (see Annex R), which treats boundaries of GOBs with non-empty headers or slices as picture boundaries, can be used to avoid spatial error propagation due to motion compensation across the boundaries of the GOBs or slices when this mode is applied to a smaller unit than a picture, such as a GOB or slice. The information to signal which picture is selected for prediction is included in the encoded bitstream.

The strategy used by the encoder to select the picture or pictures to be used for prediction is out of the scope of this Recommendation.

U.3 Forward-channel syntax

The syntax is altered in the picture, Group of Blocks (GOB), and slice layers. When indicated by a parameter MRPA being equal to "1", the syntax is also altered in the macroblock layer. In the picture, GOB, and slice layers, an Enhanced Reference Picture Selection layer (ERPS layer) is inserted. In the macroblock layer, picture reference parameters are inserted under certain conditions to enable multi-picture motion compensation.

U.3.1 Syntax of the picture, GOB, and slice layer

The Enhanced Reference Picture Selection syntax for the PLUS header (otherwise as shown in Figure 8) is shown in Figure U.2. The fields of RPSMF, PN, and the ERPS layer are inserted into the PLUS header. The fields of TRPI, TRP, BCI, and BCM are not present (since they are only needed for the RPS mode of Annex N, which is not allowed when the ERPS mode is active).

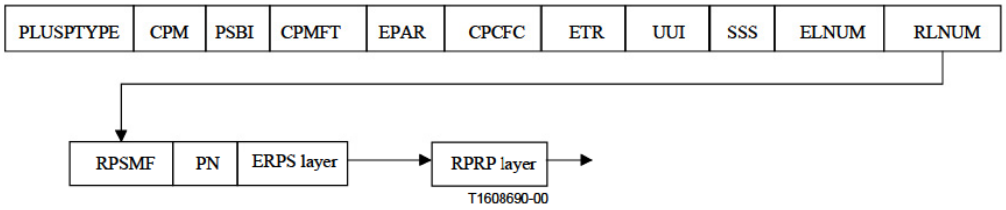


Figure U.2/H.263 – Structure of PLUS header for the ERPS mode

The syntax for the GOB layer is shown in Figure U.3. The fields of PNI, PN, NOERPSL, and the ERPS layer are added to the syntax (otherwise defined as in Figure 9).

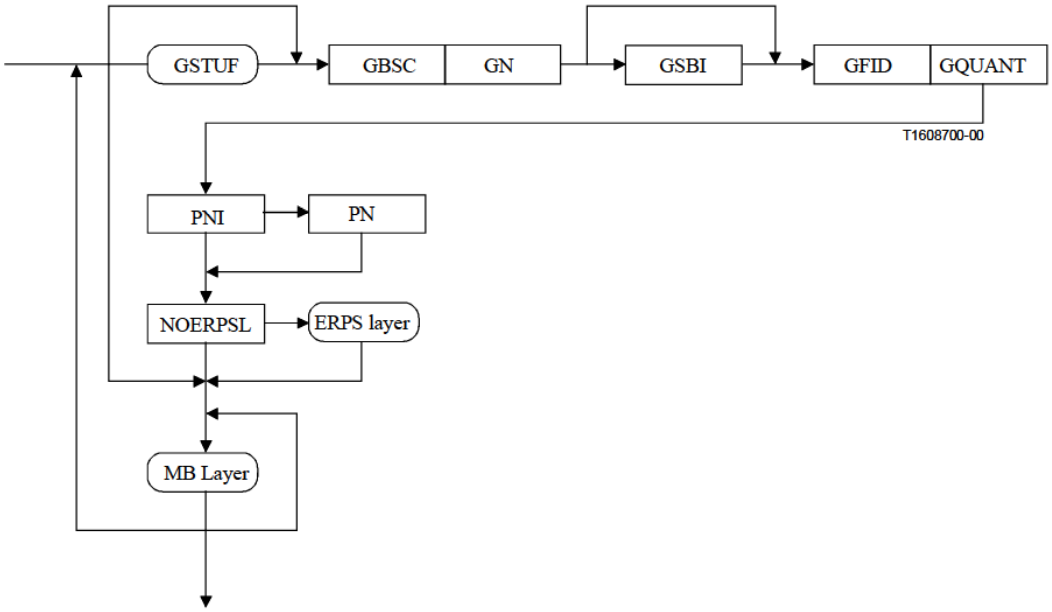


Figure U.3/H.263 – Structure of GOB layer for the ERPS mode

When the optional Slice Structured mode (see Annex K) is in use, the syntax of the slice layer is modified in the same way as the GOB layer. The syntax for the slice layer is shown in Figure U.4. The slice that immediately follows the picture start code in the bitstream also includes all of the added fields PNI, PN, NOERPSL, and the ERPS layer.

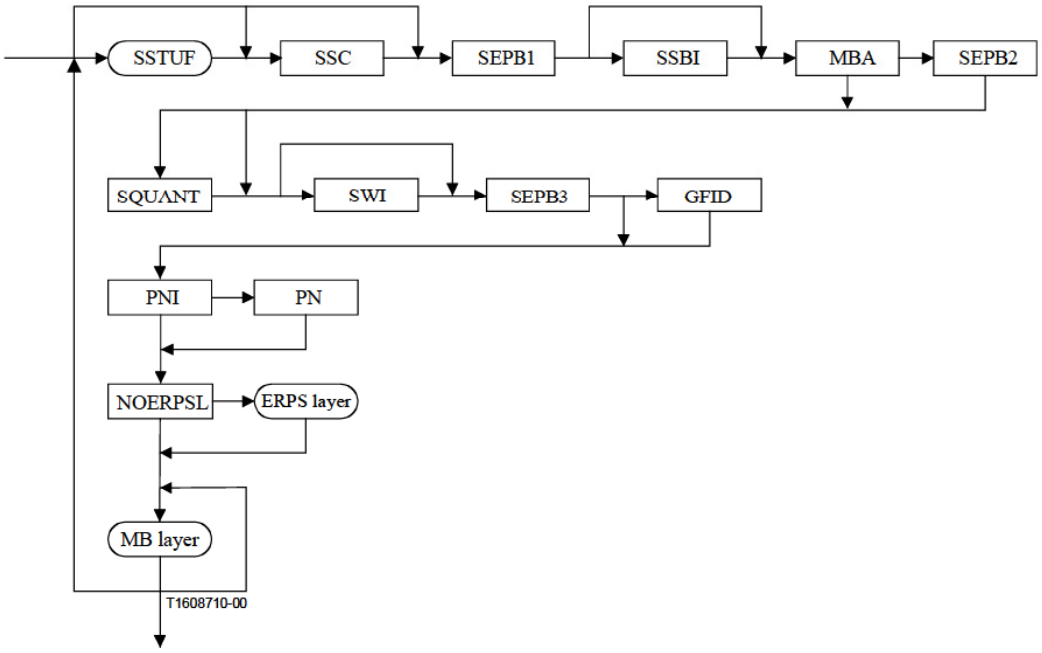


Figure U.4/H.263 – Structure of slice layer for the ERPS mode

The ERPS layer is shown in Figure U.5.

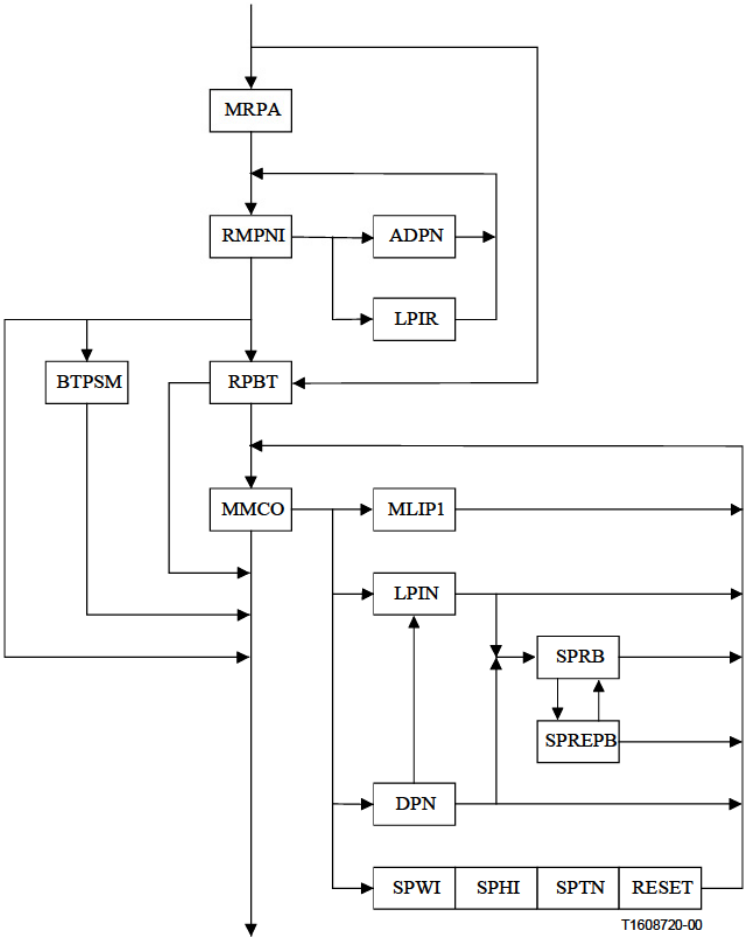


Figure U.5/H.263 – Structure of the ERPS layer

Variable length codes for the ADPN, LPIR, MLIP1, DPN, LPIN, SPTN, PR, PR_0, PR_2, PR_3, PR_4, PRB and PRFW fields are given in Table U.1.

Table U.1/H.263 – Variable length codes for ADPN, LPIR, MLIP1, DPN, LPIN, SPTN, PR, PR_0, PR_2, PR_3, PR_4, PRB, and PRFW

Absolute position	Number of bits	Codes
0	1	1
"x_0"+1 (1:2)	3	0$x_0$0
"x_1x_0"+3 (3:6)	5	0$x_1$1$x_0$0
"$x_2x_1x_0$"+7 (7:14)	7	0$x_2$1$x_1$1$x_0$0
"$x_3x_2x_1x_0$"+15 (15:30)	9	0$x_3$1$x_2$1$x_1$1$x_0$0
"$x_4x_3x_2x_1x_0$"+31 (31:62)	11	0$x_4$1$x_3$1$x_2$1$x_1$1$x_0$0
"$x_5x_4x_3x_2x_1x_0$"+63 (63:126)	13	0$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$0
"$x_6x_5x_4x_3x_2x_1x_0$"+127 (127:254)	15	0$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$0
"$x_7x_6x_5x_4x_3x_2x_1x_0$"+255 (255:510)	17	0$x_7$1$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$0
"$x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+511 (511:1022)	19	0$x_8$1$x_7$1$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$0
"$x_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+1023 (1023:2046)	21	0$x_9$1$x_8$1$x_7$1$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$0
"$x_{10}x_9x_8x_7x_6x_5x_4x_3x_2x_1x_0$"+2047 (2047:4094)	23	0x_{10}1$x_9$1$x_8$1$x_7$1$x_6$1$x_5$1$x_4$1$x_3$1$x_2$1$x_1$1$x_0$0

U.3.1.1 Reference Picture Selection Mode Flags (RPSMF) (3 bits)

RPSMF is a 3-bit fixed-length codeword that is present in the PLUS header whenever the ERPS mode is in use (regardless of the value of UFEP). RPSMF shall not be present in the GOB or slice layer. When present, RPSMF indicates which type of back-channel messages are needed by the encoder. The values of RPSMF shall be as defined in 5.1.13.

U.3.1.2 Picture Number Indicator (PNI) (1 bit)

PNI is a single-bit fixed-length codeword that is always present at the GOB or slice layer when the ERPS mode is in use, and is not present in the PLUS header. When present, PNI indicates whether or not the following PN field is also present.

"0": PN field is not present.

"1": PN field is present.

U.3.1.3 Picture Number (PN) (10 bits)

PN is a 10-bit fixed-length codeword that is always present in the PLUS header when the ERPS mode is in use, and is present at the GOB or slice layer only when indicated by PNI.

PN shall be incremented by 1 for each coded and transmitted picture, in a 10-bit modulo operation, relative to the PN of the previous stored picture. The term "stored picture" is defined in U.3.1.5.7. For EI and EP pictures, PN shall be incremented from the value in the last stored EI or EP picture within the same scalability enhancement layer. For B pictures, PN shall be incremented from the value in the most temporally-recent stored non-B picture in the reference layer of the B picture which precedes the B picture in bitstream order (a picture which is temporally subsequent to the B picture). B pictures are not stored in the multi-picture buffer, as they are not used as references for subsequent pictures. Thus, a picture immediately following a B picture in the reference layer of the B picture or another B picture which immediately follows a B picture shall have the same PN as the

B picture. Similarly, if a non-B picture is present in the bitstream which is not stored, the picture following this non-B picture (in the same enhancement layer, in the case of Annex O operation) shall have the same PN as the non-stored non-B picture.

In a usage scenario known as "Video Redundancy Coding", the ERPS mode may be used by some encoders in a manner in which more than one representation is sent for the pictured scene at the same temporal instant (usually using different reference pictures). In such a case in which the ERPS mode is in use and in which adjacent pictures in the bitstream have the same temporal reference and the same picture number, the decoder shall regard this occurrence as an indication that redundant copies have been sent of approximately the same pictured scene content, and shall decode and use the first such received picture while discarding the subsequent redundant picture(s).

The PN serves as a unique ID for each picture stored in the picture buffer within 1024 coded and stored pictures. Therefore, a picture cannot be kept in the buffer after more than 1023 subsequent coded and stored pictures (in the same enhancement layer, in the case of Annex O operation) unless it has been assigned a long-term picture index as specified below. The encoder shall ensure that the bitstream shall not specify retaining any short-term picture after more than 1023 subsequent stored pictures. A decoder which encounters a picture number on a current stored picture having a value equal to the picture number of some other short-term picture in the multi-picture buffer should treat this condition as an error.

U.3.1.4 No Enhanced Reference Picture Selection Layer (NOERPSL) (1 bit)

NOERPSL is a single-bit fixed-length codeword that is present at the GOB or slice level whenever the ERPS mode is in use. It is not present in the PLUS header. The values of NOERPSL shall be as follows:

"0": The ERPS layer is sent.

"1": The ERPS layer is not sent.

If NOERPSL is "1", all ERPS settings and re-mappings in effect for the picture shall be applied also for the relevant video picture segment that follows the GOB or slice layer data. ERPS layer information sent at the GOB or slice level governs the decoding process for the video picture segment preceded by the GOB or slice level data, and does not affect the decoding process of any other video picture segment. (See Annex R for the definition of a video picture segment.)

U.3.1.5 Enhanced Reference Picture Selection layer (ERPS) (variable length)

The ERPS layer is always present at the picture level when the ERPS mode is in use, and is present at the GOB or slice level if NOERPSL is "0". It specifies the buffer indexing used to decode the current picture or video picture segment, and manages the contents of the picture buffer.

U.3.1.5.1 Multiple Reference Pictures Active (MRPA) (1 bit)

MRPA is a single-bit fixed-length codeword that is present only if the picture coding type indicates a P picture, an EP picture, an Improved PB frame, or B picture. MRPA is the first element in the ERPS layer if present. MRPA specifies whether the number of active reference pictures for forward-prediction or backward-prediction decoding of the current picture or video picture segment may be larger than one. The value of MRPA shall be as follows:

"1": More than one reference picture may be used for forward or backward motion compensation.

"0": Only one reference picture is used for forward or backward motion compensation. In this case, the extensions of the macroblock layer syntax in U.3.2 do not apply.

MRPA may be changed from video picture segment to video picture segment, so that different video picture segments may address different numbers of reference pictures.

MRPA shall be "0" in any picture which invokes the Reference Picture Resampling mode (see Annex P), and the same picture shall be indicated as the forward reference picture to be used at both the picture and GOB or slice levels for any such current picture. If the current picture is a B picture, the backward reference picture shall have the same size as the current picture, and any reference picture resampling process shall be applied only to the forward reference picture. Reference picture resampling shall be invoked only if the multi-picture buffer contains sufficient "unused" capacity to store the resampled forward reference picture, but after the resampled reference picture is used for the decoding of the current picture, the resampled forward reference picture shall not be stored in the multi-picture buffer.

U.3.1.5.2 Re-Mapping of Picture Numbers Indicator (RMPNI) (variable length)

RMPNI is a variable-length codeword that is present in the ERPS layer if the picture is a P, EP, Improved PB, or B picture. RMPNI indicates whether any default picture indices are to be re-mapped for motion compensation of the current picture or video picture segment – and how the re-mapping of the relative indices into the multi-picture buffer is to be specified if indicated. RMPNI is transmitted using Table U.2. If RMPNI indicates the presence of an ADPN or LPIR field, an additional RMPNI field immediately follows the ADPN or LPIR field.

Table U.2/H.263 – RMPNI operations for re-mapping of reference pictures

Value	Re-mapping specified
"1"	ADPN field is present and corresponds to a negative difference to add to a picture number prediction value
"010"	ADPN field is present and corresponds to a positive difference to add to a picture number prediction value
"011"	LPIR field is present and specifies the long-term index for a reference picture
"001"	End loop for re-mapping of picture relative indexing default order

A picture reference parameter is a relative index into the ordered set of pictures. The RMPNI, ADPN, and LPIR fields allow the order of that relative indexing into the multi-picture buffer to be temporarily altered from the default index order for the decoding of a particular picture or video picture segment. The default index order is for the short-term pictures (i.e., pictures which have not been given a long-term index) to precede the long-term pictures in the reference indexing order. Within the set of short-term pictures, the default order is for the pictures to be ordered starting with the most recent buffered reference picture and proceeding through to the oldest reference picture (i.e., in decreasing order of picture number in the absence of wrapping of the ten-bit picture number field). Within the set of long-term pictures, the default order is for the pictures to be ordered starting with the picture with the smallest long-term index and proceeding up to the picture with long-term index equal to the most recent value of MLIP1 – 1.

For example, if the buffer contains three short-term pictures with short-term picture numbers 300, 302, and 303 (which were transmitted in increasing picture-number order) and two long-term pictures with long-term picture indices 0 and 3, the default index order is:

- default relative index 0 refers to the short-term picture with picture number 303;
- default relative index 1 refers to the short-term picture with picture number 302;
- default relative index 2 refers to the short-term picture with picture number 300;
- default relative index 3 refers to the long-term picture with long-term picture index 0; and
- default relative index 4 refers to the long-term picture with long-term picture index 3.

The first ADPN or LPIR field that is received (if any) moves a specified picture out of the default order to the relative index of zero. The second such field moves a specified picture to the relative

index of one, etc. The set of remaining pictures not moved to the front of the relative indexing order in this manner shall retain their default order amongst themselves and shall follow the pictures that have been moved to the front of the buffer in relative indexing order.

If MRPA is "0", no more than one ADPN or LPIR field shall be present in the same ERPS layer unless the current picture is a B picture. If the current picture is a B picture and MRPA is "0", no more than two ADPN or LPIR fields shall be present in the same ERPS layer.

Any re-mapping of picture numbers specified for some picture shall not affect the decoding process for any other picture. Any re-mapping of picture numbers specified for some video picture segment shall not affect the decoding process for any other video picture segment. A re-mapping of picture numbers specified for a picture shall only affect the decoding process for any video picture segment within that picture in two ways:

- If NOERPSL is "1" at the GOB or slice level, then the re-mapping specified at the picture level is also used for the associated video picture segment.
- If the picture is a B picture, the re-mapping specified at the picture level shall specify the calculation of the value of TR_B and TR_D for direct bidirectional prediction.

An RMPNI "end loop" indication is the last element of the ERPS layer for a B picture if MRPA is "0". In a B picture with MRPA equal to "1", an RMPNI "end loop" indication is followed by BTPSM. In a P or EP picture or Improved PB frame, an RMPNI "end loop" indication is followed by RPBT.

Within one ERPS layer, RMPNI shall not specify the placement of any individual reference picture into more than one re-mapped position in relative index order.

U.3.1.5.3 Absolute Difference of Picture Numbers (ADPN) (variable length)

ADPN is a variable-length codeword that is present only if indicated by RMPNI. ADPN follows RMPNI when present. ADPN is transmitted using Table U.1, where the index into the table corresponds to ADPN – 1. ADPN represents the absolute difference between the picture number of the currently re-mapped picture and the prediction value for that picture number. If no previous ADPN fields have been sent within the current ERPS layer, the prediction value shall be the picture number of the current picture. If some previous ADPN field has been sent, the prediction value shall be the picture number of the last picture that was re-mapped using ADPN.

If the picture number prediction is denoted PNP, and the picture number in question is denoted PNQ, the decoder shall determine PNQ from PNP and ADPN in a manner mathematically equivalent to the following:

```
if (RMPNI == "1") {     // a negative difference
  if (PNP - ADPN < 0)
    PNQ = PNP - ADPN + 1024;
  else
    PNQ = PNP - ADPN;
}else{              // a positive difference
  if (PNP + ADPN > 1023)
    PNQ  = PNP + ADPN - 1024;
  else
    PNQ  = PNP + ADPN;
}
```

The encoder shall control RMPNI and ADPN such that the decoded value of ADPN shall not be greater than or equal to 1024.

As an example implementation, the encoder may use the following process to determine values of ADPN and RMPNI to specify a re-mapped picture number in question, PNQ:

```
DELTA = PNQ - PNP;
if (DELTA < 0) {
  if (DELTA < -511)
    MDELTA = DELTA + 1024;
  else
    MDELTA = DELTA;
}else{
  if(DELTA > 512)
    MDELTA = DELTA - 1024;
  else
    MDELTA = DELTA;
}

ADPN = abs(MDELTA);
```

where abs() indicates an absolute value operation. Note that the index into Table U.1 corresponds to the value of ADPN – 1, rather than the value of ADPN itself.

RMPNI would then be determined by the sign of MDELTA.

U.3.1.5.4 Long-term Picture Index for Re-Mapping (LPIR) (variable length)

LPIR is a variable-length codeword that is present only if indicated by RMPNI. LPIR follows RMPNI when present. LPIR is transmitted using Table U.1. It represents the long-term picture index to be re-mapped. The prediction value used by any subsequent ADPN re-mappings is not affected by LPIR.

U.3.1.5.5 B-picture Two-Picture Prediction Sub-Mode (BTPSM) (1 bit)

BTPSM is a single-bit fixed-length codeword that is present only in a B picture (see Annex O) and only when MRPA is "1". It follows an RMPNI "end loop" indication and is the last element of the ERPS layer for the B picture when present. It indicates whether the two-picture backward-prediction sub-mode is in use for the picture as follows:

"0": Single-picture backward prediction.

"1": Two-picture backward prediction.

BTPSM has an implied value of "0" if not present (when MRPA is "0").

The set of pictures available for use as forward-prediction references is the set of pictures in the multi-picture buffer other than the set of backward reference pictures. The set of backward reference pictures is determined by the value of BTPSM. If single-picture backward prediction is specified by BTPSM, the first picture in (possibly re-mapped) relative index order is the only backward reference picture. If two-picture backward prediction is specified by BTPSM, the first two pictures in (possibly re-mapped) relative index order are the two backward reference pictures. The relative index for forward prediction then becomes a relative index into the set of forward reference pictures.

The contents of the multi-picture buffer are not affected by the presence of a B picture. The B picture is not stored in the multi-picture buffer and is not used as a reference for the coding of subsequent pictures.

U.3.1.5.6 Reference Picture Buffering Type (RPBT) (1 bit)

RPBT is a single-bit fixed-length codeword that specifies the buffering type of the currently decoded picture. It follows an RMPNI "end loop" indication when the picture is not an I, EI, or B picture. It is the first element of the ERPS layer if the picture is an I or EI picture. It is not present if the picture is a B picture. The values for RPBT are defined as follows:

"1": Sliding Window.

"0": Adaptive Memory Control.

In the "Sliding Window" buffering type, the current decoded picture shall be added to the buffer with index 0, and any marking of pictures as "unused" in the buffer is performed automatically in a first-in-first-out fashion among the set of short-term pictures. In this case, if the buffer has sufficient "unused" capacity to store the current picture, no additional pictures shall be marked as "unused" in the buffer. If the buffer does not have sufficient "unused" capacity to store the current picture, the picture (or pictures as necessary to free the needed amount of memory in the case of sub-picture removal) with the largest default index (or indices as necessary in the case of sub-picture removal) among the short-term pictures in the buffer shall be marked as "unused". In the "Sliding Window" buffering type, no additional information is transmitted to control the buffer contents.

In the "Adaptive Memory Control" buffering type, the encoder explicitly specifies any addition to the buffer or marking of data as "unused" in the buffer, and may also assign long-term indices to short-term pictures. The current picture and other pictures may be explicitly marked as "unused" in the buffer, as specified by the encoder. This buffering type requires further information that is controlled by memory management control operation (MMCO) parameters.

RPBT, if present in GOB or slice layers, shall be the same as in the picture layer. Any MMCO command present in GOB or slice layers shall convey the same operation as some MMCO command in the picture layer.

If the picture is a B picture, RPBT shall not be present and the decoded picture shall not be stored in the multi-picture buffer. This ensures that a B picture shall not affect the contents of the multi-picture buffer.

Similarly, the B-picture part of an Improved PB frame shall not be stored in the buffer. All control fields associated with controlling the storage of an Improved PB frame shall be considered to be associated with controlling the storage of only the P-picture part of the Improved PB frame.

U.3.1.5.7 Memory Management Control Operation (MMCO) (variable length)

MMCO is a variable-length codeword that is present only when RPBT indicates "Adaptive Memory Control", and may occur multiple times if present. It specifies a control operation to be applied to manage the multi-picture buffer memory. The MMCO parameter is followed by data necessary for the operation specified by the value of MMCO, and then an additional MMCO parameter follows – until the MMCO value indicates the end of the list of such operations. MMCO commands do not affect the buffer contents or the decoding process for the decoding of the current picture – rather, they specify the necessary buffer status for the decoding of subsequent pictures in the bitstream. The values and control operations associated with MMCO are defined in Table U.3.

Table U.3/H.263 – Memory Management Control Operation (MMCO) values

Value	Memory Management Control Operation	Associated data fields following
"1"	End MMCO loop	None (end of ERPS layer)
"011"	Mark a short-term picture as "Unused"	DPN
"0100"	Mark a long-term picture as "Unused"	LPIN
"0101"	Assign a long-term index to a picture	DPN and LPIN
"00100"	Mark short-term sub-picture areas as "Unused"	DPN and SPRB
"00101"	Mark long-term sub-picture areas as "Unused"	LPIN and SPRB
"00110"	Specify the maximum long-term picture index	MLIP1
"00111"	Specify the buffer size and structure	SPWI, SPHI, SPTN, and RESET

All memory management control operations specified using MMCO shall be specified in the picture layer. Some or all of the same operations as are specified at the picture layer may also be specified at the GOB or slice layer (with the same associated data). MMCO shall not specify memory operations at the GOB or slice layer that are not also specified with the same associated data at the picture layer.

A buffer size and structure specification MMCO command shall be the first MMCO command if present. No more than one buffer size and structure specification MMCO command shall be present in a given ERPS layer. A buffer size and structure specification MMCO command with RESET equal to "1" shall be present in the first picture in which the ERPS mode is activated in any series of ERPS mode pictures in the bitstream. A buffer size and structure specification MMCO command with RESET equal to "1" shall precede any use of MMCO to indicate marking sub-picture areas of any short-term or long-term pictures as "unused". The sub-picture width and height specified in a buffer size and structure specification MMCO command shall not differ from the value of these parameters in a prior buffer size and structure specification MMCO command unless the current picture is an I or EI picture with RESET equal to "1". The picture height and width shall not change within the bitstream except within a picture containing a buffer size and structure specification MMCO command with RESET equal to "1" (or within a picture in which the ERPS mode is not in use).

If a B picture using single-picture backward prediction is present in the bitstream, exactly one temporally subsequent non-B picture in the reference layer of the B picture shall precede the B picture in bitstream order, as specified in O.2. No memory management control operations shall be present within any ERPS layer of this immediately temporally subsequent non-B picture within the reference layer of the B picture which mark any part of that immediately temporally succeeding non-B picture as "unused", since that reference layer picture is needed for display until after the decoding of the B picture.

The transmission order constraints specified in O.2 are adjusted as necessary for B pictures using two-picture backward prediction. If a B picture using two-picture backward prediction is present in the bitstream, exactly two temporally subsequent non-B pictures in the reference layer of the B picture shall precede the B picture in bitstream order. The other restrictions on the transmission order of the B picture in the bitstream specified in O.2 shall apply, but as adjusted for the use of two temporally subsequent reference layer pictures. No memory management control operations shall be present within any ERPS layer of these two immediately temporally subsequent two non-B pictures within the reference layer of the B picture which mark any part of these two non-B pictures as "unused", since these reference layer pictures are needed for display until after the decoding of the B picture.

A "stored picture" is defined as a non-B picture which does not contain an MMCO command in its ERPS layer which marks that picture as "unused". If the current picture is not a stored picture, its ERPS layer shall not contain any of the following types of MMCO commands:

- an MMCO command to specify the buffer size and structure with RESET equal to "1";
- any MMCO command which marks any other picture as "unused" that has not also been marked as "unused" in the ERPS layer of a prior stored picture;
- any MMCO command which assigns a long-term index to a picture that has not also been assigned the same long-term index in the ERPS layer of a prior stored picture; or
- any MMCO command which marks sub-picture areas of any picture as "unused" that have not also been marked as "unused" in the ERPS layer of a prior stored picture.

U.3.1.5.8 Difference of Picture Numbers (DPN) (variable length)

DPN is present when indicated by MMCO. DPN follows MMCO if present. DPN is transmitted using codewords in Table U.1 and is used to calculate the PN of a picture for a memory control operation. It is used in order to assign a long-term index to a picture, mark a short-term picture as "unused", or mark sub-picture areas of a short-term picture as "unused". If the current decoded picture number is PNC and the decoded value from Table U.1 is DPN, an operation mathematically equivalent to the following equations shall be used for calculation of PNQ, the specified picture number in question:

```
if (PNC – DPN < 0)
  PNQ = PNC – DPN + 1024;
else
  PNQ = PNC – DPN;
```

Similarly, the encoder may compute the DPN value to encode using the following relation:

```
if (PNC – PNQ < 0)
  DPN = PNC – PNQ + 1024;
else
  DPN = PNC – PNQ;
```

For example, if the decoded value of DPN is zero and MMCO indicates marking a short-term picture as "unused", the current decoded picture shall be marked as "unused".

U.3.1.5.9 Long-term Picture Index (LPIN) (variable length)

LPIN is present when indicated by MMCO. LPIN is transmitted using codewords in Table U.1 and specifies the long-term picture index of a picture. It follows DPN if the operation is to assign a long-term index to a picture. It follows MMCO if the operation is to mark a long-term picture as "unused" or to mark sub-picture areas of a long-term picture as "unused".

U.3.1.5.10 Sub-Picture Removal Bit-map (SPRB) (fixed length)

SPRB is a fixed-length codeword that contains one bit for each sub-picture and is present when indicated by MMCO. The number of bits of SPRB data is determined by the most recent values of SPWI and SPHI. SPRB is used to indicate which sub-picture areas of a buffered picture are to be marked as "unused". SPRB follows DPN if the operation is to mark sub-picture areas of a short-term picture as "unused", and follows LPIN if the operation is to mark sub-picture areas of a long-term picture as "unused".

Sub-pictures are numbered in raster scan order starting from the upper-left corner of the picture. For example, consider a case in which a reference picture, specified by DPN, is partitioned into six sub-pictures. Let "$s_1\ s_2\ s_3\ s_4\ s_5\ s_6$" represent six bits of SPRB data. If bit s_i is "1", then the decoder should mark the ith sub-picture in the indicated reference picture as "unused". For example, if the SPRB is "000110", then the fourth and fifth sub-pictures areas are marked as "unused".

To prevent start code emulation, all necessary SPREPB emulation prevention bits shall be inserted within or following the SPRB data as specified in U.3.1.5.11.

If SPRB is present and the specified picture has been previously been affected by a prior SPRB bit-map, the bit-map specified by SPRB shall contain a "1" for any sub-picture area that contained a "1" in the previous SPRB bit-map. Every SPRB bit-map shall contain at least one bit having the value "0" and at least one bit having the value "1".

U.3.1.5.11 Sub-Picture Removal Emulation Prevention Bit (SPREPB) (one bit)

SPREPB is a single-bit fixed-length codeword having the value "1" which shall be inserted immediately after any string of 8 consecutive zero bits of SPRB data.

U.3.1.5.12 Maximum Long-term Picture Index Plus 1 (MLIP1) (variable length)

MLIP1 is a variable-length codeword that is present if indicated by MMCO. MLIP1 follows MMCO if present. MLIP1 is transmitted using codewords in Table U.1. If present, MLIP1 is used to determine the maximum index allowed for long-term reference pictures (until receipt of another value of MLIP1). The decoder shall initially assume MLIP1 is "0" until some other value has been received. Upon receiving an MLIP1 parameter, the decoder shall consider all long-term pictures having indices greater than the decoded value of MLIP1 – 1 as "unused" for referencing by the decoding process for subsequent pictures. For all other pictures in the multi-picture buffer, no change of status shall be indicated by MLIP1.

U.3.1.5.13 Sub-Picture Width Indication (SPWI) (7 bits)

SPWI is a fixed-length codeword of 7 bits that is present if indicated by MMCO. SPWI follows MMCO when indicated. SPWI specifies the width of a sub-picture in units of 16 luminance samples, such that the indicated sub-picture width is $16 \cdot (\text{SPWI} + 1)$ luminance samples. The current picture has a width in sub-picture units of ceil(ceil(pw/16)/(SPWI + 1)) sub-pictures, where pw is the width of the picture and "/" indicates floating-point division. For positive numbers, the ceiling function, ceil(x), equals x if x is an integer and otherwise ceil(x) equals one plus the integer part of x. If a minimum picture unit (MPU) size defining the minimum width and height of a sub-picture has been negotiated by external means (for example, ITU-T Rec. H.245), the sub-picture width specified by SPWI shall be an integer multiple of the width of the MPU; otherwise, the sub-picture width specified by SPWI shall be such that SPWI is equal to ceil(pw/16) – 1.

U.3.1.5.14 Sub-Picture Height Indication (SPHI) (7 bits)

SPHI is a fixed-length codeword of 7 bits that is present if SPWI is present (as indicated by MMCO). SPHI follows SPWI if present. SPHI specifies the height of a sub-picture in units of 16 luminance samples, such that the indicated sub-picture height is $16 \cdot \text{SPHI}$. The allowed range of values of SPHI is from 1 to 72. The current picture has a height of ceil(ceil(ph/16)/SPHI) sub-pictures, where ph is the height of the picture and "/" indicates floating-point division. If a minimum picture unit (MPU) size defining the minimum width and height of a sub-picture has been negotiated by external means (for example, ITU-T Rec. H.245), the sub-picture height specified by SPHI shall be an integer multiple of the height of the MPU; otherwise, the sub-picture height specified by SPHI shall be such that SPHI is equal to ceil(ph/16).

U.3.1.5.15 Sub-Picture Total Number (SPTN) (variable length)

SPTN is a variable-length codeword that is present if SPWI and SPHI are present (as indicated by MMCO). SPTN follows SPHI if present. SPTN is coded using Table U.1, where the index into Table U.1 corresponds to the decoded value of SPTN – 1. The decoded value of STPN is the total operational size capacity of the multi-picture buffer in units of sub-pictures as specified by SPWI and SPHI. The memory capacity needed for the decoding of current pictures is not included in SPTN – only the memory capacity needed for storing the reference pictures to use for the prediction of other pictures. When sub-picture removal is not in use (i.e., when SPWI and SPHI have whole-picture dimensions), the maximum number of active short-term reference pictures (for example, for sliding window operation) is thus given by SPTN minus the number of pictures that have been assigned to long-term indices and have not been subsequently marked as "unused".

U.3.1.5.16 Buffer Reset Indicator (RESET) (1 bit)

RESET is a single-bit fixed-length codeword that is present if SPWI, SPHI, and SPTN are present (as indicated by MMCO). RESET follows SPTN if present. The values of RESET shall be as follows:

"0": The buffer contents are not reset.

"1": The buffer contents are reset.

If RESET is "1", all pictures in the multi-picture buffer (but not the current picture unless specified separately) shall be marked "unused" (including both short-term and long-term pictures).

U.3.2 Macroblock layer syntax

U.3.2.1 P-picture and Improved PB frames macroblock syntax

The macroblock layer syntax is modified if the ERPS layer is present for P pictures and Improved PB frames when the number of selected forward reference pictures may be greater than one, as indicated by MRPA. The field MRPA is signalled in the ERPS layer. The macroblock layer syntax is shown in Figure U.6 when MRPA is "1". Otherwise, the macroblock syntax format in a P picture or Improved PB frame is not altered from that shown in Figure 10.

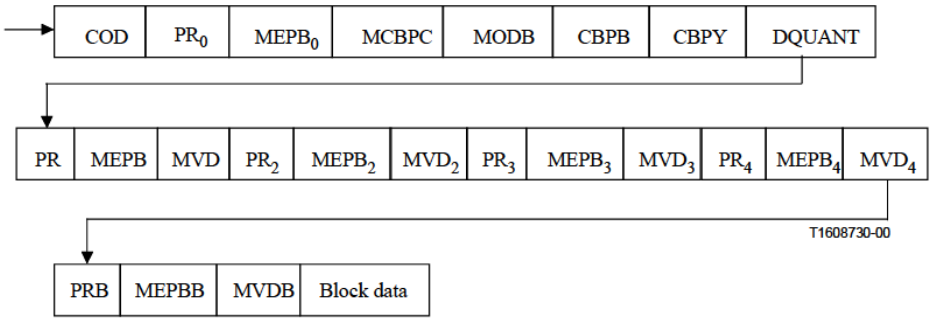


Figure U.6/H.263 – Structure of P-picture and Improved PB frame macroblock layer for the ERPS mode

U.3.2.1.1 Interpretation of COD

If the COD bit is "1", no further information is transmitted for the macroblock. In that case, the decoder shall treat the macroblock as an INTER macroblock with the motion vector for the entire macroblock equal to zero, picture reference parameter equal to zero, and with no coefficient data. If the COD bit is "0", indicating that the macroblock is coded, the syntax of the macroblock layer is depicted in Figure U.6 with the fields PR_0, PR, PR_2, PR_3, PR_4, and PRB being included in the syntax. PR_0, PR, PR_2, PR_3, PR_4, and PRB each consist of a variable length codeword as given in Table U.1.

U.3.2.1.2 Picture Reference Parameter 0 (PR_0) (variable length)

PR_0 is a variable length codeword as specified in Table U.1. It is present whenever COD is "0". If PR_0 has a decoded value of zero (codeword "1"), it indicates that further information will follow for the macroblock. If decoded as non-zero, it indicates the coding of the macroblock using only a picture reference parameter.

If the field PR_0 does not have a decoded value of zero (codeword "1"), no further information is transmitted for this macroblock. In that case the decoder shall treat the macroblock as an INTER macroblock with the motion vector for the whole block equal to zero, the picture reference parameter equal to PR_0, and with no coefficient data.

If the field PR_0 has a decoded value of zero (codeword "1"), the macroblock is coded. The meaning and usage of the fields MCBPC, CBPB, CBPY, and DQUANT remains unaltered. The field PR is included together with the field MVD for all INTER macroblocks (and in Improved PB frames mode also for INTRA macroblocks). The use of MODB in Improved PB frames is described in U.3.2.1.4.

U.3.2.1.3 Macroblock Emulation Prevention Bit 0 ($MEPB_0$) (1 bit)

$MEPB_0$ is a single-bit fixed-length codeword having the value "1" that follows PR_0 if, and only if, PR_0 is present and has a decoded value of "1" (codeword "000"), and either of the following two conditions are satisfied:

1) the slice structured mode (see Annex K) is in use; or
2) the COD for the current macroblock immediately follows after another macroblock which also has COD = "0" and PR_0 = "1" (codeword "000"), and the PR_0 of the previous macroblock is not followed by an $MEPB_0$ bit.

The purpose of $MEPB_0$ is to prevent start-code emulation and, in the slice structured mode, to aid in determining the number of macroblocks in a slice.

U.3.2.1.4 Macroblock Picture Reference parameters (PR, $PR_{2\text{-}4}$, and PRB) (variable length)

PR is the primary picture reference parameter. PR is present whenever MVD is present. The three codewords $PR_{2\text{-}4}$ are included together with $MVD_{2\text{-}4}$ if indicated by PTYPE and if MCBPC specifies an INTER4V or INTER4V+Q macroblock (a macroblock of type 2 or 5 in Tables 8 and 9). $PR_{2\text{-}4}$ and $MVD_{2\text{-}4}$ are only present when in Advanced Prediction mode (see Annex F) or Deblocking Filter mode (see Annex J). PRB is only present in an Improved PB frame when MODB indicates that MVDB is present. PR, $PR_{2\text{-}4}$, and PRB each specify a picture reference relative index into the multi-picture buffer.

PR is used as the picture reference parameter for motion compensation of the entire macroblock if the macroblock is not an INTER4V or INTER4V+Q macroblock. If the macroblock is an INTER4V or INTER4V+Q macroblock, PR is used for motion compensated prediction of the first of the four 8×8 luminance blocks in the macroblock and for the two chrominance blocks of the macroblock (with the motion compensation process otherwise as specified in 6.1). $PR_{2\text{-}4}$ are used for motion compensation of the remaining three 8×8 blocks of luminance data in the macroblock. If MODB indicates that MVDB is present, PRB is the picture reference parameter for forward prediction of the B part of the Improved PB frame.

In Improved PB frames when MODB indicates B_{PB} bidirectional prediction, the values of TR_D and TR_B shall be computed as the temporal reference increments based on the temporal reference data of the current picture and that of the most recent previous reference picture, regardless of whether or not the most recent previous reference picture has been re-mapped to a difference relative index order, marked as "unused", or assigned to a long-term index. The picture used as the forward reference picture for B_{PB} bidirectional prediction in Improved PB frames shall be the picture specified by PR.

U.3.2.1.5 Macroblock Emulation Prevention Bits (MEPB, $MEPB_{2\text{-}4}$ and MEPBB) (1 bit each)

MEPB, $MEPB_{2\text{-}4}$, and MEPBB are each a single bit having the value "1" if present. Each shall be present if, and only if, the Unrestricted Motion Vector mode (see Annex D) is not in use and the associated PR, $PR_{2\text{-}4}$, or PRB field is present and has the decoded value "1" (codeword "000"). Their purpose is to prevent start-code emulation.

U.3.2.2 B-picture and EP-picture macroblock syntax

The macroblock layer syntax for B and EP pictures (see Annex O) is modified in a similar fashion as in P pictures. The COD bit, if equal to "1", indicates a skipped macroblock as defined in Annex O, using a picture reference parameter of zero for the forward (skipped) prediction in an EP picture and for the forward part of direct (skipped) bidirectional prediction in a B picture and using the first backward prediction picture for the backward part of direct (skipped) bidirectional

prediction in a B picture (in the case of two-picture backward prediction, as when BSBBW is present and equal to "0"). If COD is "0", a PR_0 parameter is inserted into the syntax and is used in a similar manner as described in U.3.2.1.2. If PR_0 is present and does not have a decoded value of zero (codeword "1"), it indicates that the macroblock is to be predicted with forward INTER prediction using a zero-valued motion vector and a picture reference parameter of PR_0. If PR_0 has a decoded value of zero, MBTYPE follows and specifies the macroblock type. The format of the CBPC, CBPY, and DQUANT fields is unchanged. The MVDFW and MVDBW fields are encoded in the same manner as when the ERPS mode is not in use, but are each used in conjunction with a picture reference, and possibly an emulation prevention bit.

For a B picture, the backward reference pictures in the multi-picture buffer are defined as follows:

- in the case of single-picture backward prediction, there is only one backward reference picture, which is the first picture in (possibly re-mapped) relative index order; and
- in the case of two-picture backward prediction, there are two backward reference pictures, which are the first two pictures in (possibly re-mapped) relative index order.

The forward reference pictures in the multi-picture buffer are defined as the pictures in the multi-picture buffer other than the backward reference pictures. The relative indexing for forward prediction is a relative index into the forward reference picture set, and the relative indexing for backward prediction is a relative index into the backward reference picture set.

For example, if the buffer contains three short-term pictures with short-term picture numbers 300, 302, and 303 (which were transmitted in increasing picture-number order) and two long-term pictures with long-term picture indices 0 and 3, the default index order in the case of two-picture backward prediction is:

- default backward relative index 0 refers to the short-term picture with picture number 303;
- default backward relative index 1 refers to the short-term picture with picture number 302;
- default forward relative index 0 refers to the short-term picture with picture number 300;
- default forward relative index 1 refers to the long-term picture with long-term picture index 0; and
- default forward relative index 2 refers to the long-term picture with long-term picture index 3;

and in the case of single-picture backward prediction:

- the single default backward reference picture is the short-term picture with picture number 303;
- default forward relative index 0 refers to the short-term picture with picture number 302;
- default forward relative index 1 refers to the short-term picture with picture number 300;
- default forward relative index 2 refers to the long-term picture with long-term picture index 0; and
- default forward relative index 3 refers to the long-term picture with long-term picture index 3;

and if these pictures have been re-mapped to a new relative indexing order of short-term picture 302, followed by short-term picture 303, followed by long-term picture 0, followed by short-term picture 300, followed by long-term picture 3, the new relative index order in the case of two-picture backward prediction is:

- re-mapped backward relative index 0 refers to the short-term picture with picture number 302;
- re-mapped backward relative index 1 refers to the short-term picture with picture number 303;

- re-mapped forward relative index 0 refers to the long-term picture with long-term picture index 0;
- re-mapped forward relative index 1 refers to the short-term picture with picture number 300; and
- re-mapped forward relative index 2 refers to the long-term picture with long-term picture index 3;

and in the case of single-picture backward prediction:

- the single re-mapped backward reference picture is the short-term picture with picture number 302;
- re-mapped forward relative index 0 refers to the short-term picture with picture number 303;
- re-mapped forward relative index 1 refers to the long-term picture with long-term picture index 0;
- re-mapped forward relative index 2 refers to the short-term picture with picture number 300; and
- re-mapped forward relative index 3 refers to the long-term picture with long-term picture index 3.

The TR_D used for direct bidirectional prediction in a B picture shall be computed as the temporal reference increment between the first forward reference picture in (possibly re-mapped) relative index order and the first backward reference picture in (possibly re-mapped) relative index order (i.e., if two-picture backward prediction is in use, this would be the picture referenced when BSBBW is "0" as described in U.3.2.2.3). The TR_B used for direct bidirectional prediction in a B picture shall be computed as the temporal reference increment between the B picture and the first forward reference picture in (possibly re-mapped) relative index order. The relative index order used in the computation of TR_D and TR_B shall be that specified by the ERPS layer at the picture level of the B picture syntax (i.e., re-mappings at the GOB or slice level shall not affect the values of TR_D and TR_B). (See Figure U.7.)

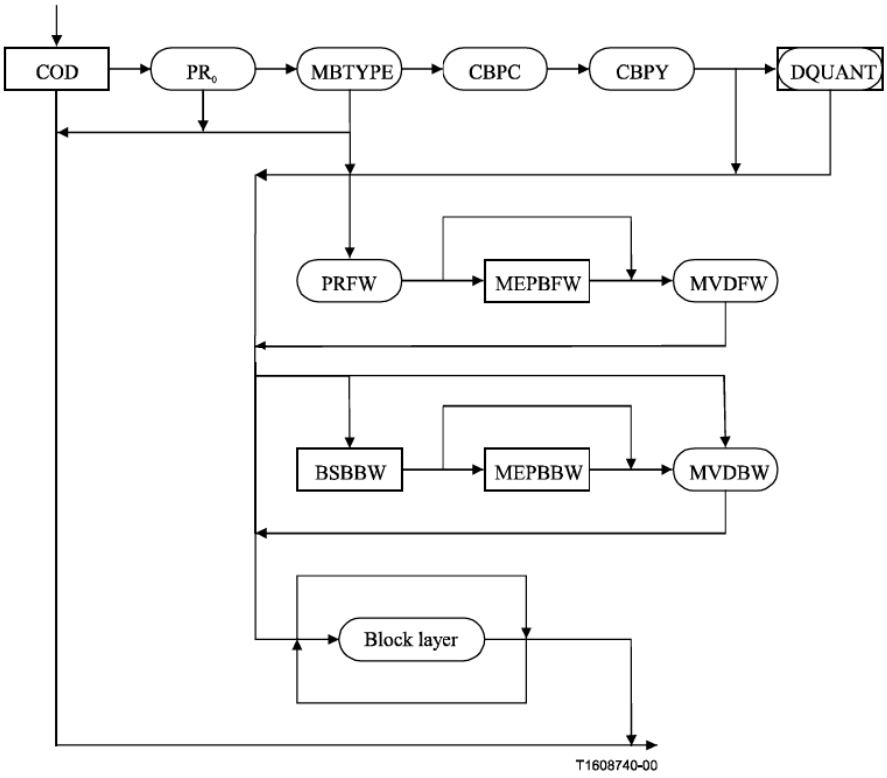


Figure U.7/H.263 – Structure of EP- and B-picture macroblock layer for the ERPS mode

U.3.2.2.1 Picture Reference for Forward Prediction (PRFW) (variable length)

PRFW is a variable-length picture reference parameter that is present whenever forward motion vector data is present, and is encoded using Table U.1. PRFW is a relative index into the set of forward reference pictures.

U.3.2.2.2 Emulation Prevention Bit for Forward Prediction (MEPBFW) (1 bit)

MEPBFW is a single-bit fixed-length codeword having the value "1" which shall be inserted after PRFW if and only if PRFW is present and has a decoded value of "1" (codeword "000") and the unrestricted motion vector mode (see Annex D) is not in use.

U.3.2.2.3 B-picture Selection Bit for Backward Prediction (BSBBW) (1 bit)

BSBBW is a single-bit fixed-length codeword that is present only for B pictures when MVDBW is present and only when two-picture backward prediction is specified for the B-picture operation. The meaning of this bit shall be defined as:

"0": Prediction from the first backward reference picture in relative index order (in default order, this would be the most recent short-term reference picture if that picture has not been assigned a long-term index or marked as "unused").

"1": Prediction from the second backward reference picture in relative index order (in default order, this would be the second-most recent short-term reference picture if neither of the last two reference pictures has been assigned a long-term index or marked as "unused").

U.3.2.2.4 Emulation Prevention Bit for Backward Prediction (MEPBBW) (1 bit)

MEPBBW is a single-bit fixed-length codeword having the value "1" that is present only under the following conditions:

- BSBBW is present and equal to "0"; and
- the unrestricted motion vector mode (see Annex D) is not in use; and
- BSBBW is preceded by five bits having the value "00000".

U.4 Decoder process

The decoder for the ERPS mode stores the reference pictures for inter-picture decoding in a multi-picture buffer. The decoder may need additional memory capacity to store the multiple decoded pictures (relative to the memory capacity needed without support of the ERPS mode). The decoder replicates the multi-picture buffer of the encoder according to the reference picture buffering type and any memory management control operations specified in the bitstream. The buffering scheme may also be operated when partially erroneous pictures are decoded.

Each transmitted and stored picture is assigned a Picture Number (PN) which is stored with the picture in the multi-picture buffer. PN represents a sequential picture counting identifier for stored pictures. PN is constrained, using modulo 1024 arithmetic operation. For the first transmitted picture, PN should be "0". For each and every other transmitted and stored picture, PN shall be increased by 1 (within a given scalability layer, if Annex O is in use). If the difference (modulo 1024) of the PNs of two consecutively received and stored pictures is not 1, the decoder should infer a loss of pictures or corruption of data. In such a case, a back-channel message indicating the loss of pictures may be sent to the encoder.

Besides the PN, each picture stored in the multi-picture buffer has an associated index, called the default index. When a picture is first added to the multi-picture buffer, it is given default index 0 – unless it is assigned to a long-term index. The indices of pictures in the multi-picture buffer are modified when pictures are added to or removed from the multi-picture buffer.

The pictures stored in the multi-picture buffers can also be divided into two categories: long-term pictures and short-term pictures. A long-term picture can stay in the multi-picture buffer for a long time (more than 1023 coded and stored picture intervals). The current picture is initially considered a short-term picture. Any short-term picture can be changed to a long-term picture by assigning it a long-term index according to information in the bitstream. The PN is the unique ID for all short-term pictures in the multi-picture buffer. When a short-term picture is changed to a long-term picture, it is also assigned a long-term picture index (LPIN). A long-term picture index is assigned to a picture by associating its PN to an LPIN. Once a long-term picture index has been assigned to a picture, the only potential subsequent use of the long-term picture's PN within the bitstream shall be in a repetition of the long-term index assignment. The PNs of the long-term pictures are unique within 1024 transmitted and stored pictures. Therefore, the PN of a long-term picture cannot be used for assignment of a long-term index after 1023 transmitted subsequent stored pictures. LPIN becomes the unique ID for the life of a long-term picture.

PN (for a short-term picture) or LPIN (for a long-term picture) can be used to re-map the pictures into re-mapped indices for efficient reference picture addressing.

U.4.1 Decoder process for short-/long-term picture management

The decoder may have both long-term pictures and short-term pictures in its multi-picture buffer. The MLIP1 field is used to indicate the maximum long-term picture index allowed in the buffer. If no prior value of MLIP1 has been sent, no long-term pictures shall be in use, i.e., MLIP1 shall initially have an implied value of "0" upon invocation of the ERPS mode. Upon receiving an MLIP1 parameter, a new MLIP1 shall take effect until another value of MLIP1 is received. Upon receiving a new MLIP1 parameter in the bitstream, all long-term pictures with associated long-term indices greater than or equal to MLIP1 shall be considered marked "unused". The frequency of transmitting MLIP1 is out of the scope of this Recommendation. However, the encoder should send an MLIP1 parameter upon receiving an error message, such as an INTRA request message.

A short-term picture can be changed to a long-term picture by using an MMCO command with an associated DPN and LPIN. The short-term picture number is derived from DPN and the long-term picture index is LPIN. Upon receiving such an MMCO command, the decoder shall change the short-term picture with PN indicated by DPN to a long-term picture and shall assign it to the long-term index indicated by LPIN. If a long-term picture with the same long-term index already exists in the buffer, the previously-existing long-term picture shall be marked "unused". An encoder shall not assign a long-term index greater than MLIP1 – 1 to any picture. If LPIN is greater than MLIP1 – 1, this condition should be treated by the decoder as an error. For error resilience, the encoder may send the same long-term index assignment operation or MLIP1 specification message repeatedly. If the picture specified in a long-term assignment operation is already associated with the required LPIN, no action shall be taken by the decoder. An encoder shall not assign the same picture to more than one long-term index value. If the picture specified in a long-term index assignment operation is already associated with a different long-term index, this condition should be treated as an error. An encoder shall only change a short-term picture to a long-term picture within 1024 transmitted consecutive stored pictures. In other words, a short-term picture shall not stay in the short-term buffer after more than 1023 subsequent stored pictures have been transmitted. An encoder shall not assign a long-term index to a short-term picture that has been marked as "unused" by the decoding process prior to the first such assignment message in the bitstream. An encoder shall not assign a long-term index to a picture number that has not been sent.

U.4.2 Decoder process for reference picture buffer mapping

The decoder employs indices when referencing a picture for motion compensation on the macroblock layer using the fields PR_0, PR, PR_2, PR_3, PR_4, PRB, PRFW, and BSBBW. In pictures other than B pictures, these indices are the default relative indices of pictures in the multi-picture buffer when the fields ADPN and LPIR are not present in the current picture, GOB, or slice layer as applicable, and are re-mapped indices when these fields are present. In B pictures, the first one or two pictures (depending on BTPSM) in relative index order are used for backward prediction, and the forward picture reference parameters specify a relative index into the remaining pictures for use in forward prediction.

The indices of pictures in the multi-picture buffer can be re-mapped onto newly specified indices by transmitting the RMPNI, ADPN, and LPIR fields. RMPNI indicates whether ADPN or LPIR is present. If ADPN is present, RMPNI specifies the sign of the difference to be added to a picture number prediction value. The ADPN value corresponds to the absolute difference between the PN of the picture to be re-mapped and a prediction of that PN value. The first transmitted ADPN is computed as the absolute difference between the PN of the current picture and the PN of the picture to be re-mapped. The next transmitted ADPN field represents the difference between the PN of the previous picture that was re-mapped using ADPN and that of another picture to be re-mapped. The process continues until all necessary re-mapping is complete. The presence of re-mappings specified using LPIR does not affect the prediction value for subsequent re-mappings using ADPN. If RMPNI indicates the presence of an LPIR field, the re-mapped picture corresponds to a long-term picture with a long-term index of LPIR. If any pictures are not re-mapped to a specific order by

RMPNI, these remaining pictures shall follow after any pictures having a re-mapped order in the indexing scheme, following the default order amongst these non-re-mapped pictures.

If the decoder detects a missing picture, it may invoke some concealment process, and may insert an error-concealed picture into the multi-picture buffer. Missing pictures can be identified if one or several picture numbers are missing or if a picture not stored in the multi-picture buffer is indicated in a transmitted ADPN or LPIR. Concealment may be conducted by copying the closest temporally preceding picture that is available in the multi-picture buffer into the position of the missing picture. The temporal order of the short-term pictures in the multi-picture buffer can be inferred from their default index order and PN fields. In addition or instead, the decoder may send a forced INTRA update signal to the encoder by external means (for example, ITU-T Rec. H.245), or the decoder may use external means or back-channel messages (for example, ITU-T Rec. H.245) to indicate the loss of pictures to the encoder. A concealed picture may be inserted into the multi-picture buffer when using the "Sliding Window" buffering type. If a missing picture is detected when decoding a GOB or Slice layer, the concealment may be applied to the picture as if the missing picture had been detected at the picture layer.

U.4.3 Decoder process for sub-picture removal

Sub-picture removal may be used to reduce the amount of memory required to save multiple reference pictures. In sub-picture removal, each reference picture is partitioned into smaller equal-sized sub-pictures. The memory reduction is accomplished by marking undesired sub-pictures as "unused". The strategy used by the encoder to decide which of the sub-pictures to mark as "unused" is outside the scope of this Recommendation. The encoder signals to the decoder the size of the sub-pictures and which of the sub-pictures to mark as "unused" using MMCO commands in the enhanced reference picture selection (ERPS) layer. The encoder shall not send information in the bitstream that causes any samples in reference pictures or sub-pictures that it has caused to be marked as "unused" to be indicated for use in the prediction of subsequent pictures.

The sub-picture removal capability is negotiated by external means (for example, ITU-T Rec. H.245). In addition, the decoder signals, also by external means, the minimum partition unit (MPU) which is described in terms of a minimum width and height (in units of 16 luminance samples) of a sub-picture and the total amount of memory it has available for its multi-picture buffer. Memory management is facilitated by the partition rules described below.

Each reference picture is partitioned into rectangular sub-pictures of equal size. The encoder specifies the sub-picture size which shall be an integer multiple of the MPU. The width and height of the sub-picture shall be integer multiples of the minimum width and height negotiated externally as the MPU. The upper-left-hand corner of the first sub-picture is coincident with the upper-left-hand corner of the reference picture. Consequently, the entire partition may be described by specifying the width and height of a sub-picture. If the picture size is not an integer multiple of the sub-picture size, some sub-pictures may extend beyond the right and bottom boundaries of the reference picture. When a sub-picture that extends past the reference picture boundary is saved, a convenient memory management strategy is to set aside enough memory to save the entire sub-picture, rather than just the memory necessary to save the portion of the reference picture that lies within that sub-picture. This is the convention which shall be followed in any calculation of buffer spare capacity for the purpose of determining buffer fullness (e.g., in order to determine whether to automatically mark buffered pictures as "unused" in "sliding window" operation). A decoder designed such that each sub-picture occupies the same amount of memory will prevent the possibility of memory fragmentation.

An example method designed to access referenced picture samples when sub-picture removal is in use is described briefly as follows. One important element in any reference picture access technique is a mechanism to identify where the samples in each sub-picture are stored in memory. If there are R reference pictures and each picture is partitioned into S sub-pictures, then there are a total of $K = R \cdot S$ sub-pictures. For example, the sub-picture in the upper-left hand corner of the first

reference picture number can be considered sub-picture number 0, and the sub-picture to the right of it can be considered sub-picture number 1 and so on in raster scan order progressing from reference picture 1 to R until all K sub-pictures have a label. The total buffer capacity is SPTN sub-picture memory buffers, and SPTN is typically less than K. A K-element array can be defined, subPicMem[K], such that t = subPicMem[k] corresponds to the sub-picture memory area that contains the samples in the kth sub-picture. For example, a case can be considered in which R = 5 reference pictures each have S = 12 sub-pictures. Then the samples for the 6th sub-picture in reference picture 3 would be found in sub-picture memory area t = subPicMem[k] where $k = 3 \cdot S + 6 = 42$.

For example, when referencing samples for motion-compensated prediction of one block of luminance or chrominance data when the Advanced Prediction and Reduced Resolution Update optional modes are not in use, it is necessary to acquire $n \times m$ samples, where n and m may take values of 8 or 9 to accommodate half-integer motion compensation. Since the samples in one block may lie in up to four different sub-pictures, four separate cases must be considered. In all cases, the first step is to find the location in memory that contains the upper-left hand sample (U) of the block to be referenced. The sub-picture containing U can be identified by dividing the horizontal or vertical location of U by the sub-picture width or height. If U lies in sub-picture k, then that sample will be located in the subPicMem[k] sub-picture memory area. Next, if both the sample $m - 1$ samples to the right of U (i.e., the upper-right-hand corner of the block) and the sample $n - 1$ samples down from U (i.e., the lower-left-hand corner of the block) lie in sub-picture k, this can be considered case number one. If the sample $n - 1$ samples down from U lies within k, but the sample $m - 1$ samples to the right of U does not, this can be considered case two. If the sample $m - 1$ samples to the right of U lies within k, but the sample $n - 1$ down does not, this can be considered case three. Otherwise, when both the sample $m - 1$ samples to the right of U and the one $n - 1$ samples down lie outside of sub-picture k, this can be considered case four.

In case number one, all samples in the reference block are contained within the kth sub-picture. In this case, all relevant $n \times m$ samples may be found in sub-picture memory area subPicMem[k] and it is a simple task to access them. In case two, the samples that lie in the kth sub-picture can be obtained from sub-picture memory area subPicMem[k] and the remaining samples can be obtained from subPicMem[k_r] where k_r is the sub-picture to the right of k. In case three, the samples that lie in the kth sub-picture can be obtained from memory area subPicMem[k], and the remaining samples can be obtained from subPicMem[k_d] where k_d is the sub-picture below k. In case four, the samples that lie in the kth sub-picture can be obtained from sub-picture memory area subPicMem[k] and the remaining samples can be obtained from memory areas subPicMem[k_r], subPicMem[k_d] and subPicMem[k_{rd}] where k_r and k_d are defined above and k_{rd} is the sub-picture to the right and below k.

U.4.4 Decoder process for multi-picture motion compensation

Multi-picture motion compensation is applied if the MRPA field indicates the use of more than one reference picture. For multi-picture motion compensation, the decoder chooses a reference picture as indicated using the fields PR_0, PR, PR_2, PR_3, PR_4, PRB, PRFW and BSBBW on the macroblock layer. Once the reference picture is specified, the decoding process for motion compensation proceeds as described in 6.1.

In case four motion vectors per macroblock are used and the MRPA field indicates the use of more than one reference picture, the picture reference index for both chrominance blocks is that associated with the first of the four motion vectors (with the motion compensation process otherwise as specified by 6.1).

U.4.5 Decoder process for reference picture buffering

The buffering of the currently decoded picture can be specified using the reference picture buffering type (RPBT) for non-B pictures. The buffering may follow a first-in, first-out ("Sliding Window") mode. Alternatively, the buffering may follow a customized adaptive buffering ("Adaptive Memory Control") operation that is specified by the encoder in the forward channel. B pictures do not affect buffer contents.

The "Sliding Window" buffering type operates as follows. First, the decoder determine whether the picture can be stored into "unused" buffer capacity. If there is insufficient "unused" buffer capacity, the short-term picture with the largest default index (i.e., the oldest short-term picture in the buffer) shall be marked as "unused". This process is repeated if necessary (in the case of sub-picture removal) until sufficient memory capacity is freed to hold the current decoded picture. The current picture is stored in the buffer and assigned a default relative buffer index of zero. The default relative index of all other short-term pictures is incremented by one. The default relative index of all long-term pictures is incremented by one minus the number of short-term pictures removed.

In the "Adaptive Memory Control" buffering type, specified pictures or sub-picture areas may be removed from the multi-picture buffer explicitly. The currently decoded picture, which is initially considered a short-term picture, may be inserted into the buffer with default relative index 0, may be assigned to a long-term index, or may be marked as "unused" by the encoder. Other short-term pictures may also be assigned to long-term indices. The buffering process shall operate in a manner functionally equivalent to the following: First, the current picture is added to the multi-picture buffer with default relative index 0, and the default relative indices of all other pictures is incremented by one. Then, the MMCO commands are processed:

- If MMCO indicates a reset of the buffer contents by using RESET equal to "1", all pictures in the buffer are marked as "unused" except the current picture (which will be the picture with default relative index 0 since a buffer reset must be the first MMCO command as required by U.3.1.5.7).
- If MMCO indicates a maximum long-term index using MLIP1, all long-term pictures having long-term indices greater than or equal to MLIP1 are marked as "unused" and the default relative index order of the remaining pictures are not affected.
- If MMCO indicates that a picture is to be marked as "unused" in the multi-picture buffer and if that picture has not already been marked as "unused", the specified picture is marked as "unused" in the multi-picture buffer and the default relative index of all subsequent pictures in default order is decremented by one.
- If MMCO indicates that sub-picture areas of some picture are to be marked as "unused" in the multi-picture buffer, the specified sub-picture areas are marked as "unused" and the default relative index order of the pictures is not affected. As required by U.3.1.5.10, not all sub-picture areas of any given picture will be marked "unused" by a sub-picture removal MMCO command (instead, the encoder should send an MMCO command marking the picture as a whole as "unused").
- If MMCO indicates the assignment of a long-term index to a specified short-term picture and if the specified long-term index has not already been assigned to the specified short-term picture, the specified short-term picture is marked in the buffer as a long-term picture with the specified long-term index. If another picture is already present in the buffer with the same long-term index as the specified long-term index, the other picture is marked as "unused". All short-term pictures that were subsequent to the specified short-term picture in default relative index order and all long-term pictures having a long-term index less than the specified long-term index have their associated default relative indices decremented by one. The specified picture is assigned to a default relative index of one plus the highest of the decremented default relative indices, or zero if there are no such decremented indices.

The resulting buffered quantity of pictures or sub-picture regions not marked as "unused" shall not exceed the buffer capacity indicated by the most recent value of SPTN. If the decoder detects this condition, it should be treated as an error.

U.5 Back-channel messages

An out-of-band channel, which need not necessarily be reliable, can be used to convey back-channel messages. The syntax of this out-of-band channel (which could be a separate logical channel, for example using ITU-T Rec. H.223 or ITU-T Rec. H.225.0) should be the one defined herein. The "videomux" operation of back-channel messages as defined in Annex N is not supported in the ERPS mode.

U.5.1 BCM separate logical channel layer

The BCM layer as specified in U.5.2 should be carried by a BCM separate logical channel layer as shown in Figure U.8.

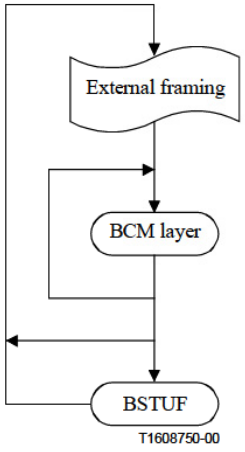


Figure U.8/H.263 – Structure of BCM separate logical channel layer for ERPS mode

U.5.1.1 External framing

External framing of back-channel messages should be provided as shown in Figure U.8. The external framing is used to determine the starting point for the back-channel messages and the amount of back-channel message data to follow.

U.5.1.2 Back-channel Stuffing (BSTUF) (variable length)

BSTUF is a variable-length codeword that may be present only after the last back-channel message in an external frame. BSTUF consists of a codeword of variable length consisting of one or more bits of value "0".

U.5.2 Back-channel message layer syntax

The syntax for the back-channel message (BCM) layer defined herein shall be as shown in Figure U.9.

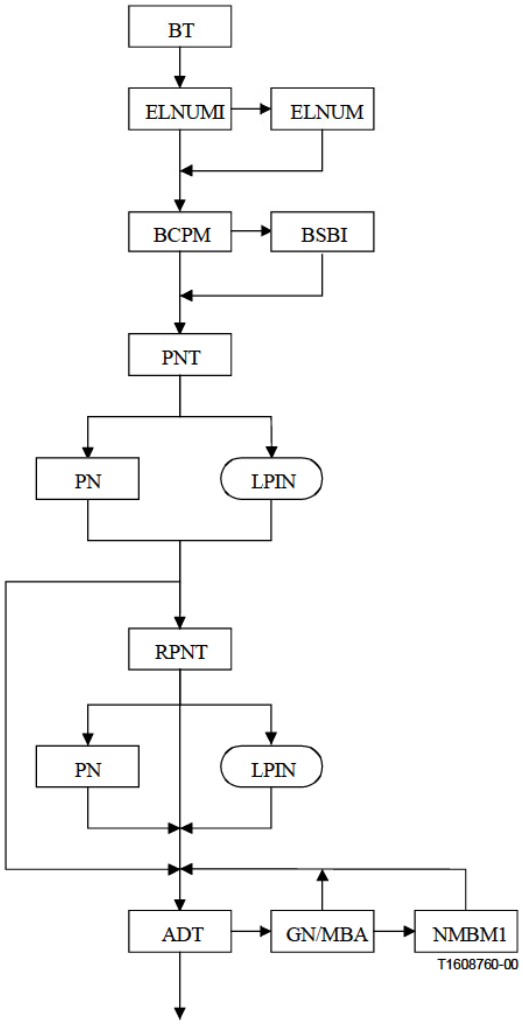


Figure U.9/H.263 – Structure of Back-Channel Message (BCM) layer for ERPS mode

U.5.2.1 Back-channel message Type (BT) (2 bits)

BT is a 2-bit fixed-length codeword which indicates the type of back-channel message. BT is the first codeword present in each back-channel message. Which type or types of message are requested by the encoder is indicated in the RPSMF field of the forward-channel syntax. The values of BT shall be defined as:

"00": Reserved for future use.

"01": Reserved for future use.

"10": NACK. This indicates the loss or erroneous decoding of the corresponding part of the forward channel data.

"11": ACK. This indicates the correct decoding of the corresponding part of the forward channel data.

U.5.2.2 Enhancement Layer Number Indication (ELNUMI) (1 bit)

ELNUMI is a single-bit fixed-length codeword that follows BT in the back-channel message. ELNUMI shall be "0" unless the optional Temporal, SNR, and Spatial Scalability mode (see Annex O) is used in the forward channel and some enhancement layers of the forward channel are combined in one logical channel and the back-channel message refers to an enhancement layer (rather than the base layer), in which case ELNUMI shall be "1".

U.5.2.3 Enhancement Layer Number (ELNUM) (4 bits)

ELNUM is a 4-bit fixed-length codeword that is present only if ELNUMI is "1". It follows ELNUMI if present. When present, ELNUM contains the layer number of the enhancement layer referred to in the back-channel message.

U.5.2.4 Back-channel CPM Indicator (BCPM) (1 bit)

BCPM is a single-bit fixed-length codeword that follows ELNUMI or ELNUM in the back-channel message. BCPM shall be "0" unless the CPM mode (see 5.2.4 and Annex C) is used in the forward channel data, in which case BCPM shall be "1". If BCPM is "1", this indicates that BSBI is present.

U.5.2.5 Back-channel Sub-Bitstream Indicator (BSBI) (2 bits)

BSBI is a 2-bit fixed-length codeword that follows BCPM when present. BSBI is present only if BCPM is "1". BSBI is the natural binary representation of the Sub-Bitstream number in the forward channel data to which the back-channel message refers (see 5.2.4 and Annex C).

U.5.2.6 Picture Number Type (PNT) (1 bit)

PNT is a single-bit fixed-length codeword that is always present and follows BCPM or BSBI in the back-channel message. The values of PNT shall be defined as:

"0": The message concerns a picture specified by a short-term picture number (PN).

"1": The message concerns a picture specified by a long-term picture index (LPIN).

PNT is followed by PN or LPIN, depending on the value of PNT. PN and LPIN shall be represented as specified for use in forward channel data in U.3.1.3 and U.3.1.5.9, respectively.

U.5.2.7 Requested Picture Number Type (RPNT) (2 bits)

RPNT is a 2-bit fixed-length codeword that is present only if BT indicates a NACK message. It follows PN or LPIN when present. It determines how to identify a picture in the multi-picture buffer which may be used as a reference for the coding of subsequent pictures. The values of RPNT shall be defined as:

"00": No valid pictures in buffer – Buffer should be reset by an I or EI picture with RESET equal to "1".

"01": No particular picture is identified to be used as a reference.

"10": A picture which may be used as a reference is identified by a short-term picture number (PN).

"11": A picture which may be used as a reference is identified by a long-term picture index (LPIN).

If RPNT is "10" or "11", RPNT is followed by PN or LPIN, depending on the value of RPNT. PN and LPIN shall be represented as specified for use in forward channel data in U.3.1.3 and U.3.1.5.9, respectively. Typically the PN or LPIN specified using RPNT identifies the last correctly decoded spatially-corresponding picture area for the picture or region identified in the back-channel message.

U.5.2.8 Additional Data Type (ADT) (2 bits)

ADT is a 2-bit fixed-length codeword that is present after PN, LPIN, or RPNT, as determined by PNT (in an ACK message) or RPNT (in a NACK message). It may occur multiple times if present. It specifies the type of additional data used to identify a region of the picture of concern to which the back-channel message applies. The values of ADT shall be defined as:

"00": End of additional data.

"01": A region is identified by only a GN/MBA field.

"10": A region is identified as a raster-scan area within a picture by GN/MBA and NMBM1.

"11": A region is identified as a raster-scan area within a rectangular slice by GN/MBA and NMBM1.

If ADT is "00", no more data follows in the back-channel message. If ADT is "01", ADT is followed by GN/MBA and then by another ADT. If ADT is "10" or "11", ADT is followed by GN/MBA and NMBM1 and then by another ADT.

If ADT is "10", the region is identified as a region starting at a particular spatial location specified by GN/MBA and containing a specified number of macroblocks in raster-scan order within the picture. If ADT is "11", the region is identified as a region starting at a particular spatial location specified by GN/MBA and containing a specified number of macroblocks in raster-scan order within a rectangular slice. If ADT is present only once and is "00", the region identified is the picture as a whole. If ADT is present more than once, the value "00" is used only to end the loop rather than to identify a region.

U.5.2.9 GOB Number/Macroblock Address (GN/MBA) (5/6/7/9/11/12/13/14 bits)

GN/MBA is a fixed-length codeword which specifies a GOB number or macroblock address. GN/MBA follows ADT when present. GN/MBA is present when indicated by ADT. If the optional Slice Structured mode (see Annex K) is not in use, GN/MBA contains the GOB number of the beginning of an area to which the back-channel message refers. If the optional Slice Structured mode is in use, GN/MBA contains the macroblock address of the beginning of the area to which the back-channel message refers. The length of this field shall be as specified elsewhere in this Recommendation for GN or MBA.

U.5.2.10 Number of Macroblocks Minus 1 (NMBM1) (5/6/7/9/11/12/13/14 bits)

NMBM1 is a fixed-length codeword which specifies a number of macroblocks. NMBM1 is present when indicated by ADT. It follows GN/MBA when present. It contains the natural representation of the number of specified macroblocks minus 1. The length of this field shall be the length defined for a macroblock address in K.2.5 and in Table K.2.

Annex V

Data-partitioned slice mode

V.1 Scope

This annex describes the optional data-partitioned slice (DPS) mode of H.263. The capability of this mode is signalled by external means (for example, ITU-T Rec. H.245). The use of this mode shall be indicated by setting the formerly-reserved bit 17 of the optional part of the PLUSPTYPE (OPPTYPE) to "1". This mode uses the header structure defined in Annex K.

Data partitioning provides robustness in error prone environments. This is accomplished using a rearrangement of the H.263 syntax to enable early detection of and recovery from errors that have been introduced during transmission.

V.2 Structure of data partitioning

When data partitioning is used, the data is arranged as a video picture segment, as defined in R.2. The MBs in the segment are rearranged so that the header information for all the MBs in the segment are transmitted together, followed by the MVs for all the MBs in the segment, and then by the DCT coefficients for all the MBs in the segment. The segment header uses the same syntax as described in K.2. The header, MV, and DCT partitions are separated by markers, allowing for resynchronization at the end of the partition in which an error occurred. Each segment shall contain the data for an integer number of MBs. When this mode is in use, the syntax shown in Figure V.1 shall be used.

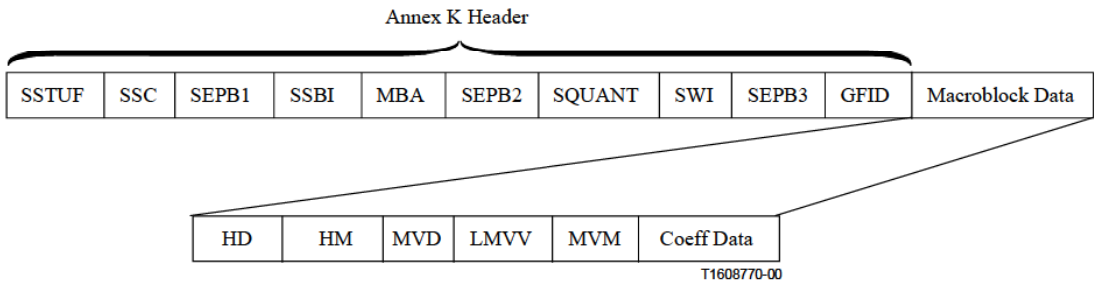


Figure V.1/H.263 – Data partitioning syntax

Note that when this annex is not active, the MV and DCT data are transmitted in an interleaved fashion for all the MBs in a video picture segment, in which case an error normally results in the loss of all information for the remaining MBs in the packet.

V.2.1 Header Data (HD) (Variable length)

The Header Data field contains the COD and MCBPC information for all the MBs in the packets, plus the MODB data in case of PB-frames or Improved PB-frames. A reversible variable length code (RVLC) is used to combine the COD and the MCBPC for all the MBs in the packet. This code is shown in Tables V.1 through V.5. If Annex O is in use, the COD is only combined with the MB TYPE to form the RVLC for B and EP pictures using Tables V.3 and V.4, and the CBPC is coded with codewords in Table O.4. If COD = 0 and Annex G or Annex M is in use, the codeword for the COD+MCBPC shall be immediately followed by the reversible variable-length encoded data corresponding to the MODB field of the macroblock. Table V.6 shall be used for PB-frames, Table V.7 shall be used for Improved PB-frames.

V.2.2 Header Marker (HM) (9 bits)

A codeword of 9 bits. Its value is 1010 0010 1. The HM terminates the header partition. When reversed decoding is used by a decoder, the decoder searches for this marker. This value cannot occur naturally in the HD field.

V.2.3 Motion vector data layer (Variable length)

V.2.3.1 Motion vector difference coding

For the motion vectors, the RVLC codewords shown in Table D.3 are used to encode the difference between the motion vector and the motion vector prediction. Note that this annex only uses the entropy coding from Annex D, but not its other aspects unless Annex D is also in use.

V.2.3.2 Prediction of motion vector values

The first motion vector in the packet is coded using a predictor value of 0 for both horizontal and vertical components, and the MVs for the subsequent coded MBs are coded predictively using the MV difference (MVD). This differs from the method otherwise used for coding the MVs in which the MVs following a skipped or INTRA MB are coded using a predictor value of 0 for both horizontal and vertical components.

Forward Direction: $MV_i = MV_{i-1} + MVD_i = MV_{i-1} + (MV_i - MV_{i-1})$

Backward Direction: $MV_{i-1} = MV_i - MVD_i = MV_i - (MV_i - MV_{i-1})$.

(MV_i and MVD_i are the *i*th MV and MV Difference in the packet respectively).

The motion vector information for the last motion vector in the packet is coded in this manner and is also coded again in the LMVV field as described below in V.2.4. This allows the decoder to independently decode the sequence of MVs using two different prediction paths:

1) in the forward direction, starting from the beginning of the motion data of the packet; and
2) in the backward direction, from the end of the motion data in a packet.

This provides robustness for better error detection and concealment.

NOTE 1 – When the DPS mode is not in use, motion vectors are predictively coded, with the prediction of the current motion vector being the median value of 3 motion vectors of neighboring locations as described in 6.1.1. Because packets in this annex are formed in a way such that the number of MBs coded in each packet is variable, using the median predictive coding method (which involves motion vectors on different rows of the frame) would prevent reversible decoding of the motion vectors in a slice. When the DPS mode is in use, a single prediction thread is formed for the MVs in the whole packet. This is shown in Figure V.2.

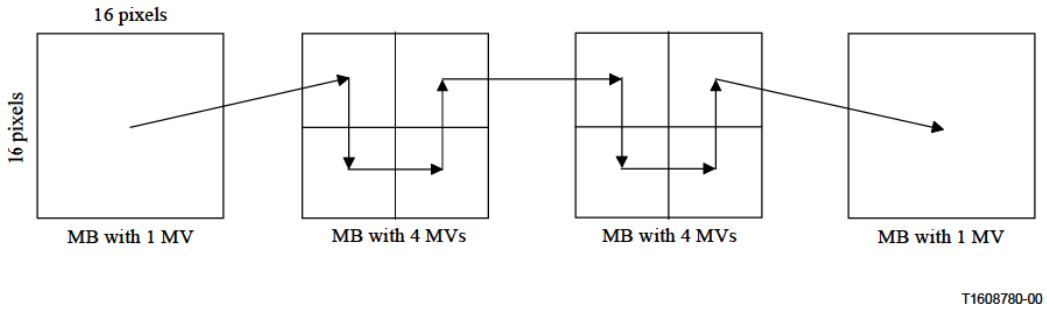


Figure V.2/H.263 – Single thread motion vector prediction

In case of B pictures or EP pictures (Annex O), MVDFW and MVDBW may be present as indicated by the MBTYPE codeword in Tables V.3 and V.4. MVDFW is predictively encoded

using the same single prediction thread as described above and MVDBW (when present in B pictures) shall be encoded as specified in O.4.6. MVDFW and MVDBW shall be coded with the codewords from Table D.3.

In case of PB-frames (Annex G) and Improved PB-frames (Annex M), the MVDB data shall be encoded as specified in corresponding annexes and shall be coded using the codewords from Table D.3.

NOTE 2 – If the backward decoding mode is engaged in a B frame (Annex O) or in Improved PB-frames (Annex M), MVDB and MVDBW should be discarded by the decoder as the Motion Vector data for the backward prediction may not be recovered properly across the packet boundaries.

V.2.3.3 Start-code emulation prevention in motion vector difference coding

The MVD start-code-emulation avoidance method is changed from the method described in D.2, in order to facilitate independent parsing in the backward direction. The MV partition shall be scanned from left to right and a MVD = 0 (codeword "1") shall be inserted after any two MVDs that are both equal to 1 (codeword "000"). If a third MVD = 1 codeword follows these two MVD = 1 codewords in the original bitstream (before insertion), it shall be considered the first MVD = 1 codeword detected in the remaining codewords in the MV partition. It shall not be considered a second MVD = 1 codeword, and shall not have a MVD = 0 codeword inserted after it. This differs from Annex D, in which the bit is only inserted when two consecutive MVD = 1 (codeword "000") form a pair (i.e., when the first MVD is the horizontal component, and the second is the vertical component). If Annex D and Annex V are both in use, this Annex V method of start-code-emulation avoidance method shall be used instead of the method described in D.2.

V.2.4 Last Motion Vector Value (LMVV) (Variable length)

The LMVV field contains the last MV in the packet. It is coded using a predictor value of 0 for both the horizontal and vertical components. If there are no motion vectors or only one motion vector in the packet, LMVV shall not be present. (This use of a fixed zero-valued predictor enables the use of reversible decoding.)

V.2.5 Motion Vector Marker (MVM) (10 bits)

A codeword of 10 bits having the value "0000 0000 01". The MVM terminates the motion vector partition. When reverse decoding is used in a decoder, the decoder searches for this marker. The Motion Vector Marker (MVM) shall not be included in the packet if the packet does not contain Motion Vector Data (if all the macroblocks in the packet are intra-coded or with CODs equal to 1).

V.2.6 Coefficient Data Layer (Variable length)

The DCT data layer contains INTRA_MODE (if present), CBPB (if present), CBPC (if present), CBPY, DQUANT (if present), and DCT coefficients coded as specified in I.2, 5.3.4, O.4.3, 5.3.5, 5.3.6, and 5.4.2, respectively. The syntax diagram of DCT Data is illustrated in Figure V.3. The presence of CBPC is indicated in Tables V.3 and V.4.

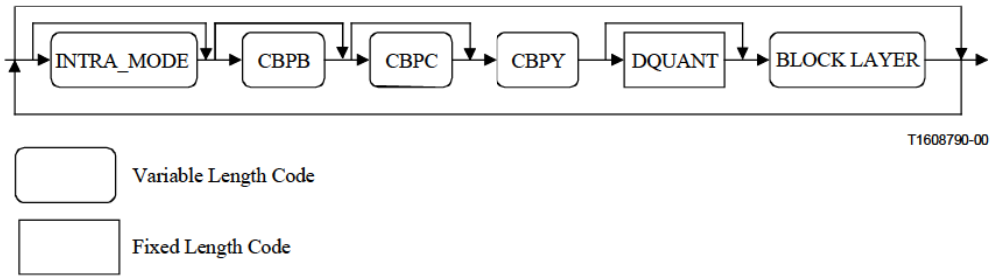


Figure V.3/H.263 – Coefficient Data syntax

V.3 Interaction with other optional modes

The DPS mode acts effectively as a sub-mode of the Slice Structured mode of Annex K, and uses its outer picture and slice header structures. The SS mode shall therefore be indicated as being in use whenever the DPS mode is in use. Both of the other sub-modes of the Slice Structured mode (the Arbitrary Slice Ordering and Rectangular Slice sub-modes) may be used in conjunction with the DPS mode.

The Syntax-Based Arithmetic Coding mode of Annex E shall not be used with this annex, as it does not allow for reversible decoding.

Annex H Forward Error Correction should not be used with this annex, as it can result in the bitstream being disrupted in undesirable places. However, the use of Annex H with the DPS mode is not forbidden, as the FEC defined in Annex H is required in some existing standard system designs.

The Temporal, SNR, and Spatial Scalability (TSSS) mode of Annex O may be used in conjunction with the DPS mode. When the TSSS and DPS modes are used together, the codewords provided in Tables V.3, V.4 and V.5 shall be used instead of those defined in Annex O.

Annex U shall not be used with this annex.

Table V.1/H.263 – COD + MCBPC RVLC table for INTRA MBs

MB type	CBPC (56)	Codeword (for combined COD + MCBPC)	Number of bits
3 (INTRA)	00	1	1
3	01	010	3
3	10	0110	4
3	11	01110	5
4 (INTRA + Q)	00	00100	5
4	01	011110	6
4	10	001100	6
4	11	0111110	7
stuffing		0011100	7

Table V.2/H.263 – COD + MCBPC RVLC table for INTER MBs

MB type	CBPC (56)	Codeword (for combined COD+MCBPC)	Number of bits
skipped		1	1
0 (INTER)	00	010	3
0	10	00100	5
0	01	011110	6
0	11	0011100	7
1 (INTER + Q)	00	01110	5
1	10	00011000	8
1	01	011111110	9
1	11	01111111110	11
2 (INTER4V)	00	0110	4
2	10	01111110	8
2	01	00111100	8
2	11	000010000	9
3 (INTRA)	00	001100	6
3	11	0001000	7
3	10	001111100	9
3	01	000111000	9
4 (INTRA + Q)	00	0111110	7
4	11	0011111100	10
4	10	0001111000	10
4	01	0000110000	10
5 (INTER4V + Q)	00	00111111100	11
5	01	00011111000	11
5	10	00001110000	11
5	11	00000100000	11
stuffing		0111111110	10

Table V.3/H.263 – MBTYPE RVLC codes for B MBs

Index	Prediction type	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
–	Direct (skipped)					1 (COD=1)	1
0	Direct			X		010	3
1	Direct + Q			X	X	001100	6
2	Forward (no texture)	X				00100	5
3	Forward	X		X		011110	6
4	Forward + Q	X		X	X	01111110	8
5	Backward (no texture)		X			0110	4
6	Backward		X	X		01110	5
7	Backward + Q		X	X	X	00111100	8
8	Bi-Dir (no texture)	X	X			0011100	7
9	Bi-Dir	X	X	X		0001000	7
10	Bi-Dir + Q	X	X	X	X	0111110	7
11	INTRA			X		00011000	8
12	INTRA + Q			X	X	011111110	9
13	Stuffing					001111100	9

Table V.4/H.263 – MBTYPE RVLC table for EP MBs

Index	Prediction type	MVDFW	MVDBW	CBPC + CBPY	DQUANT	MBTYPE	Bits
–	Forward (skipped)					1 (COD=1)	1
0	Forward	X		X		010	3
1	Forward + Q	X		X	X	0110	4
2	Upward (no texture)					01110	5
3	Upward			X		00100	5
4	Upward + Q			X	X	011110	6
5	Bi-Dir (no texture)					001100	6
6	Bi-Dir	X		X		0111110	7
7	Bi-Dir + Q	X		X	X	0011100	7
8	INTRA			X		0001000	7
9	INTRA + Q			X	X	01111110	8
10	Stuffing					00111100	8

Table V.5/H.263 – COD + MCBPC RVLC table for EI MBs

Prediction type	QCBP (56)	Codeword (for combined COD+MCBPC)	Number of bits
Upward (skipped)		1	1
0 (Upward)	00	010	3
0	01	0110	4
0	10	01110	5
0	11	00100	5
1 (Upward + Q)	00	011110	6
1	01	001100	6
1	10	0111110	7
1	11	0011100	7
2 (INTRA)	00	0001000	7
2	01	01111110	8
2	10	00111100	8
2	11	00011000	8
3 (INTRA + Q)	00	011111110	9
3	01	001111100	9
3	10	000111000	9
3	11	000010000	9
Stuffing		0111111110	10

Table V.6/H.263 – RVLC table for MODB

Index	CBPB	MVDB	Number of bits	Code
0			3	010
1		X	4	0110
2	X	X	5	01110
NOTE – "X" means that the item is present in the macroblock.				

Table V.7/H.263 – RVLC table for MODB for Improved PB-frames mode

Index	CBPB	MVDB	Number of bits	Code	Coding mode
0			3	010	Bidirectional prediction
1	X		4	0110	Bidirectional prediction
2		X	5	01110	Forward prediction
3	X	X	5	00100	Forward prediction
4			6	011110	Backward prediction
5	X		6	001100	Backward prediction
NOTE – The symbol "X" in the table above indicates that the associated syntax element is present.					

Annex W

Additional supplemental enhancement information specification

W.1 Scope

This annex describes the format of the additional supplemental enhancement information sent in the PSUPP field of the picture layer of H.263, which adds to the functionality defined in Annex L. The capability of a decoder to provide any or all of the capabilities described in this annex may be signalled by external means (for example, ITU-T Rec. H.245). Decoders which do not provide the additional capabilities may simply discard any of the newly defined PSUPP information bits that appear in the bitstream. The presence of this supplemental enhancement information is indicated by the presence of both the PEI bit, and by the following PSUPP octet whose FTYPE field has one of the two newly defined values. The basic interpretation of PEI, PSUPP, FTYPE, and DSIZE is identical to Annex L and to clauses 5.1.24 and 5.1.25.

W.2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

– ISO/IEC 10646:2003, *Information technology – Universal Multiple-Octet Coded Character Set (UCS)*.

– IETF RFC 2396 (1998), *Uniform Resource Identifiers (URI): Generic Syntax*.

W.3 Additional FTYPE values

Two values that were reserved in Annex L, Table L.1 are defined as in Table W.1.

Table W.1/H.263 – FTYPE function type values

13	Fixed-Point IDCT
14	Picture Message

W.4 Recommended maximum number of PSUPP octets

When using any of the aforementioned FTYPE functions defined in this annex, the total number of PSUPP octets per picture should, in relation to the coded picture size, be kept reasonably small, and should not exceed 256 octets regardless of the coded picture size.

NOTE – Some data transmission protocols used for conveyance of the video bitstream may provide for external repetition of picture header contents for error resilience purposes, and may place limits on the amount of such data that can be repeated from a picture header (e.g., 504 bits in the IETF RFC 2429 packetization format). The inclusion of a large number of PSUPP octets may result in the lack of such an external protocol to provide for full repetition of the picture header contents.

W.5 Fixed-point IDCT

The fixed-point IDCT function indicates that a particular IDCT approximation is used in construction of the bitstream. DSIZE shall be equal to 1 for the fixed-point IDCT function. The

octet of PSUPP data that follows specifies the particular IDCT implementation. A value of 0 indicates the reference IDCT 0 as described in W.5.3; values of 1 through 255 are reserved.

W.5.1 Decoder operation

The capability of a decoder to perform a particular fixed-point IDCT may be signalled to the encoder by external means (for example, ITU-T Rec. H.245). When receiving an encoded bitstream with the fixed-point IDCT indication, a decoder shall use the particular fixed-point IDCT if it is capable of doing so.

W.5.2 Removal of forced updating

Annex A specifies the accuracy requirements for the inverse discrete cosine transform (IDCT), allowing numerous compliant implementations. To control accumulation of errors due to mismatched IDCTs at the encoder and decoder, clause 4.4, Forced Updating, requires that macroblocks be coded in INTRA mode at least once every 132 times when coefficients are transmitted.

If the fixed-point IDCT function type is indicated in the bitstream, then the forced updating requirement is removed, and the frequency of INTRA coding is unregulated. An encoder should continue to use forced updating, however, unless it has ascertained through external means that the decoder is capable of the particular fixed-point IDCT specified herein; otherwise there may be mismatch.

W.5.3 Reference IDCT 0

The reference IDCT 0 is any implementation that, for every input block, produces identical output values as the C source program listed below.

NOTE – This fixed-point IDCT is compliant with Annex A, but is not compliant with the extended range of values requirement in Annex A of ITU-T Rec. H.262 | ISO/IEC 13818-2.

```
/*****************************************************************************
 *
 *                            FIXED-POINT IDCT
 *
 * Fixed-point fast, separable idct
 * Storage precision: 16 bits signed
 * Internal calculation precision: 32 bits signed
 * Input range: 12 bits signed, stored in 16 bits
 * Output range: [-256, +255]
 * All operations are signed
 *
 *****************************************************************************/

/*
 * Includes
 */

#include <stdlib.h>
#include <stdio.h>

/*
 * Typedefs
 */

typedef short int REGISTER;  /* 16 bits signed */
typedef long int LONG; /* 32 bits signed */

/*
 * Global constants
 */
```

```
const REGISTER cpo8   = 0x539f;     /* 32768*cos(pi/8)*1/sqrt(2)  */
const REGISTER spo8   = 0x4546;     /* 32768*sin(pi/8)*sqrt(2)    */
const REGISTER cpo16  = 0x7d8a;     /* 32768*cos(pi/16)   */
const REGISTER spo16  = 0x18f9;     /* 32768*sin(pi/16)   */
const REGISTER c3po16 = 0x6a6e;     /* 32768*cos(3*pi/16) */
const REGISTER s3po16 = 0x471d;     /* 32768*sin(3*pi/16) */
const REGISTER OoR2   = 0x5a82;     /* 32768*1/sqrt(2)    */

/*
 * Function declarations
 */

void Transpose(REGISTER block[64]);
void HalfSwap(REGISTER block[64]);
void Swap(REGISTER block[64]);
void Scale(REGISTER block[64], signed char sh);
void Round(REGISTER block[64], signed char sh,
        const REGISTER min, const REGISTER max);
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh);
void Rotate(REGISTER *x, REGISTER *y,
         signed char sha, signed char shb,
         const REGISTER a, const REGISTER b,
         int inv);
void Butterfly(REGISTER column[8], char pass);
void IDCT(REGISTER block[64]);

/*
 * Transpose():
 *      Transpose a block
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void Transpose(REGISTER block[64])
{
  int i, j;
  REGISTER temp;

  for (i=0; i<8; i++) {
    for (j=0; j<i; j++) {
      temp = block[8*i+j];
      block[8*i+j] = block[8*j+i];
      block[8*j+i] = temp;
    }
  }
  return;
}

/*
 * HalfSwap():
 *      One-dimensional swap
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void HalfSwap(REGISTER block[64])
{
```

```
  int i;
  REGISTER temp;

  for (i=0; i<8; i++) {
    temp = block[8+i];
    block[8+i] = block[32+i];
    block[32+i] = temp;
    temp = block[24+i];
    block[24+i] = block[48+i];
    block[48+i] = temp;
    temp = block[40+i];
    block[40+i] = block[56+i];
    block[56+i] = temp;
  }
  return;
}

/*
 * Swap():
 *      Swap and transpose a block
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void Swap(REGISTER block[64])
{
  HalfSwap(block);
  Transpose(block);
  HalfSwap(block);
}

/*
 * Scale():
 *      Scale a block
 * Input:
 *      REGISTER block[64]
 *      signed char sh
 * Output:
 *      block
 * Return value:
 *      none
 */
void Scale(REGISTER block[64], signed char sh)
{
  int i;

  if (sh>0) {
    for (i=0; i<64; i++)
      block[i] >>= sh;
  }
  else {
    for (i=0; i<64; i++)
      block[i] <<= -sh;
  }
}

/*
 * Round():
 *      Performs the final rounding of an 8x8 block
 * Input:
 *      REGISTER block[64]
```

```
 *      signed char sh
 *      const REGISTER min
 *      const REGISTER max
 * Output:
 *      block
 * Return value:
 *      none
 */
void Round(REGISTER block[64], signed char sh,
       const REGISTER min, const REGISTER max)
{
  int i;

  for (i=0; i<64; i++) {
    if (block[i] < 0x00007FFF - (1<<(sh-1)))
      block[i] += (1<<(sh-1));
    else
      block[i] = 0x00007FFF;
    block[i] >>= sh;
    block[i] = (block[i]<min) ? min : ((block[i]>max) ? max : block[i]);
  }
  return;
}

/*
 * Multiply():
 *      Multiply by a constant with shift
 * Input:
 *      const REGISTER a
 *      REGISTER x
 *      signed char sh
 * Output:
 *      none
 * Return value:
 *      REGISTER, the result of the multiply
 */
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh)
{
  LONG tmp;
  REGISTER reg_out;

  /* multiply */
  tmp = (LONG)a * (LONG)x;

  /* shift */
  if (sh > 0)
    tmp >>= sh;
  else
    tmp <<= -sh;

  /* rounding and saturating */
  if (tmp < 0x7FFFFFFF - 0x00007FFF)
    tmp = tmp + 0x00007FFF;
  else
    tmp = 0x7FFFFFFF;

  reg_out = (REGISTER)(tmp >>16);

  return(reg_out);
}

/*
 * Rotate():
 *      Perform rotate operation on two registers
```

```
 * Input:
 *      REGISTER *x         pointer to the 1st register
 *      REGISTER *y         pointer to the 2nd register
 *      signed char sha     shift associated with factor a
 *      signed char shb     shift associated with factor b
 *      const REGISTER a    factor a
 *      const REGISTER b    factor b
 *      int inv             1 for inverse dct, 0 for forward dct
 * Output:
 *      *x, *y
 * Return value:
 *      none
 */
void Rotate(REGISTER *x, REGISTER *y,
         signed char sha, signed char shb,
         const REGISTER a, const REGISTER b,
         int inv)
{
  LONG tmplxa, tmplya, tmplxb, tmplyb;
  LONG tmpl1, tmpl2;

  /*
   * intermediate calculation
   */

  tmplxa = (LONG)(*x) * (LONG)a;
  if (sha > 0)
    tmplxa >>= sha;
  else
    tmplxa <<= -sha;

  tmplya = (LONG)(*y) * (LONG)a;
  if (sha > 0)
    tmplya >>= sha;
  else
    tmplya <<= -sha;

  tmplxb = (LONG)(*x) * (LONG)b;
  if (shb > 0)
    tmplxb >>= shb;
  else
    tmplxb <<= -shb;

  tmplyb = (LONG)(*y) * (LONG)b;
  if (shb > 0)
    tmplyb >>= shb;
  else
    tmplyb <<= -shb;

  /*
   * rounding and rotation
   */

  if (inv) {
    tmplxa += 0x00007FFF;
    tmplxb += 0x00007FFF;

    tmpl1 = tmplxb - tmplya;
    tmpl2 = tmplxa + tmplyb;
  }
  else {
    tmplya += 0x00007FFF;
    tmplyb += 0x00007FFF;
```

```
    tmpl1 =  tmplxb + tmplya;
    tmpl2 = -tmplxa + tmplyb;
  }

  /*
   * final rounding
   */

  *x = (REGISTER) (tmpl1 >>16);
  *y = (REGISTER) (tmpl2 >>16);

  return;
}

/*
 * Butterfly():
 *      Perform 1D IDCT on a column
 * Input:
 *      REGISTER column[8]
 *      char pass
 * Output:
 *      column
 * Return value:
 *      none
 */
void Butterfly(REGISTER column[8], char pass)
{
  int i;
  REGISTER shadow_column[8];

  /*
   * For readability, we use a shadow column
   * that contains the state of column at the
   * preceding stage of the butterfly.
   */

  /*
   * Initialization
   */

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * First Phase
   */

  Rotate(column+2, column+6, pass-2, pass-1, cpo8,   spo8,   1);
  Rotate(column+1, column+7, pass-1, pass-1, cpo16,  spo16,  1);
  Rotate(column+3, column+5, pass-1, pass-1, c3po16, s3po16, 1);

  if (pass) {
    int a, tmp=column[4], b=column[0];
    a = b+tmp;
    b = b-tmp;
    column[0] = (a - ((tmp<0) ? 1 : 0)) >> 1;
    column[4] = (b - ((tmp<0) ? 1 : 0)) >> 1;
  }
  else {
    column[0] = shadow_column[0] + shadow_column[4];
    column[4] = shadow_column[0] - shadow_column[4];
  }
```

```
  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Second Phase
   */

  column[1] = shadow_column[1] - shadow_column[3];
  column[3] = shadow_column[1] + shadow_column[3];

  column[7] = shadow_column[7] - shadow_column[5];
  column[5] = shadow_column[7] + shadow_column[5];

  column[0] = shadow_column[0] + shadow_column[6];
  column[6] = shadow_column[0] - shadow_column[6];

  column[4] = shadow_column[4] + shadow_column[2];
  column[2] = shadow_column[4] - shadow_column[2];

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Third Phase
   */

  column[7] = shadow_column[7] - shadow_column[3];
  column[3] = shadow_column[7] + shadow_column[3];

  column[1] = Multiply(OoR2, shadow_column[1], -2);
  column[5] = Multiply(OoR2, shadow_column[5], -2);

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Fourth Phase
   */

  column[4] = shadow_column[4] + shadow_column[3];
  column[3] = shadow_column[4] - shadow_column[3];

  column[2] = shadow_column[2] + shadow_column[7];
  column[7] = shadow_column[2] - shadow_column[7];

  column[0] = shadow_column[0] + shadow_column[5];
  column[5] = shadow_column[0] - shadow_column[5];

  column[6] = shadow_column[6] + shadow_column[1];
  column[1] = shadow_column[6] - shadow_column[1];

  return;
}

/*
 * IDCT():
 *      Perform 2D IDCT on a block
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
```

```
void IDCT(REGISTER block[64])
{
  int i;

  Scale(block, -4);

  for (i=0; i<8; i++)
    Butterfly(block+8*i, 0);

  Transpose(block);

  for (i=0; i<8; i++)
    Butterfly(block+8*i, 1);

  Round(block, 6, -256, 255);

  Swap(block);
}
```

For informative purposes, a related forward discrete cosine transform (FDCT) implementation is shown below. This fixed-point FDCT does not form an integral part of this Recommendation.

```
/*****************************************************************************
 *
 *                            FIXED-POINT FDCT
 *
 * Fixed-point fast, separable fdct
 * Storage precision: 16 bits signed
 * Internal calculation precision: 32 bits signed
 * Input range: 9 bits signed, stored in 16 bits
 * Output range: [-2048, +2047]
 * All operations are signed
 *
 *****************************************************************************/

/*
 * Function declarations
 */

void FButterfly(REGISTER column[8]);
void FDCT(REGISTER block[64]);

/*
 * FButterfly():
 *      Perform 1D FDCT on a column
 * Input:
 *      REGISTER column[8]
 * Output:
 *      column
 * Return value:
 *      none
 */
void FButterfly(REGISTER column[8])
{
  int i;
  REGISTER shadow_column[8];

  /*
   * For readability, we use a shadow column
   * that contains the state of column at the
   * preceding stage of the butterfly.
   */
```

```
  /*
   * Initialization
   */

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * First Phase
   */

  for (i=0; i<4; i++) {
    column[i]   = shadow_column[i] + shadow_column[7-i];
    column[7-i] = shadow_column[i] - shadow_column[7-i];
  }

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Second Phase
   */

  column[0] = shadow_column[0] + shadow_column[3];
  column[3] = shadow_column[0] - shadow_column[3];

  column[1] = shadow_column[1] + shadow_column[2];
  column[2] = shadow_column[1] - shadow_column[2];

  column[4] = Multiply(OoR2, shadow_column[4], -2);
  column[7] = Multiply(OoR2, shadow_column[7], -2);

  column[6] = shadow_column[6] - shadow_column[5];
  column[5] = shadow_column[6] + shadow_column[5];

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Third Phase
   */

  column[0] = shadow_column[0] + shadow_column[1];
  column[1] = shadow_column[0] - shadow_column[1];

  column[6] = shadow_column[6] - shadow_column[4];
  column[4] = shadow_column[6] + shadow_column[4];

  column[7] = shadow_column[7] - shadow_column[5];
  column[5] = shadow_column[7] + shadow_column[5];

  for (i=0; i<8; i++)
    shadow_column[i] = column[i];

  /*
   * Fourth Phase
   */

  Rotate(column+2, column+3, -2, -1, cpo8, spo8, 0);
  Rotate(column+4, column+5, -1, -1, cpo16, spo16, 0);
  Rotate(column+6, column+7, -1, -1, c3po16, s3po16, 0);

  return;
}
```

```
/*
 * FDCT():
 *      Perform 2D FDCT on a block
 * Input:
 *      REGISTER block[64]
 * Output:
 *      block
 * Return value:
 *      none
 */
void FDCT(REGISTER block[64])
{
  int i;

  for (i=0; i<8; i++)
    FButterfly(block+8*i);

  Transpose(block);

  for (i=0; i<8; i++)
    FButterfly(block+8*i);

  Round(block, 3, -2048, 2047);

  Swap(block);
}
```

W.6 Picture message

The picture message function indicates the presence of one or more octets representing message data. The first octet of the message data is a message header with the following structure, as shown in Figure W.1.

CONT	EBIT	MTYPE

Figure W.1/H.263 – Structure of first message octet

DSIZE shall be equal to the number of octets in the message data corresponding to a picture message function, including the first octet shown in Figure W.1.

Decoders shall parse picture message data as required by basic PSUPP syntax, but decoder response to picture messages is otherwise undefined.

W.6.1 Continuation (CONT) (1 bit)

If equal to "1", CONT indicates that the message data associated with this picture message function is part of the same logical message as the message data associated with the next picture message function. If equal to "0", CONT indicates that the message data associated with this picture message function terminates the current logical message. CONT may be used, for example, to represent logical messages that span more than 14 octets.

W.6.2 End Bit Position or Track Number (EBIT) (3 bits)

For non-text picture messages, EBIT specifies the number of least significant bits that shall be ignored in the last message octet. In non-text picture messages, if CONT is "1", or if there is only one message octet (i.e., the octet in Figure W.1), EBIT shall equal "0". The number of valid message bits for a non-text picture message function excluding the CONT/EBIT/MTYPE bits is

equal to (DSIZE – 1) × 8 – EBITS. The number of valid message bits for a logical message may be greater due to continuation.

For picture message types containing text information, EBIT shall contain a text track number. The precise meaning of the text track number is not specified herein, but should indicate a particular type (e.g., language) for the text. Track number zero should be considered the default track.

W.6.3 Message Type (MTYPE) (4 bits)

MTYPE indicates the type of message. The defined types are shown in Table W.2.

Table W.2/H.263 – MTYPE message type values

0	Arbitrary Binary Data
1	Arbitrary Text
2	Copyright Text
3	Caption Text
4	Video Description Text
5	Uniform Resource Identifier Text
6	Current Picture Header Repetition
7	Previous Picture Header Repetition
8	Next Picture Header Repetition, Reliable TR
9	Next Picture Header Repetition, Unreliable TR
10	Top Interlaced Field Indication
11	Bottom Interlaced Field Indication
12	Picture Number
13	Spare Reference Pictures
14..15	Reserved

W.6.3.1 Arbitrary binary data

Arbitrary binary data is used to convey any non-ISO/IEC 10646 UTF-8 coded binary message. The interpretation of contents of the arbitrary binary data are outside the scope of this Recommendation, but should begin with some identifying pattern (e.g., a four-octet identifier code) to aid in distinguishing one type of such data from others.

W.6.3.2 Arbitrary text

Arbitrary text is used to convey a generic ISO/IEC 10646 UTF-8 coded text message. More specific text messages such as copyright information should be represented with other message types (e.g., copyright text) as appropriate.

W.6.3.3 Copyright text

Copyright text shall be used only to convey intellectual property information regarding the source or the encoded representation in the bitstream. The copyright message shall be coded according to ISO/IEC 10646 UTF-8.

W.6.3.4 Caption text

Caption text shall be used only to convey caption information associated with the current and subsequent pictures of the bitstream. The caption message shall be coded according to ISO/IEC 10646 UTF-8. The caption text shall be inserted in the bitstream as if it were to be displayed in a separate text area where new text is appended at the end of previous text and earlier text scrolled

away from the point of insertion. The Form Feed (hexadecimal "0x000C") control code shall be used to indicate clearing of the visible text area. The End of Medium (hexadecimal "0x0019") control code shall be used to indicate "caption off" status. However, this Recommendation puts no restriction on how caption text is actually displayed and stored.

W.6.3.5 Video description text

Video description text shall be used only to convey descriptive information associated with the information contents of the current bitstream. The video description shall be coded according to ISO/IEC 10646 UTF-8. The video description text shall be inserted in the bitstream as if it were to be displayed in a separate text area where new text is appended at the end of previous text and earlier text scrolled away from the point of insertion. The Form Feed (hexadecimal "0x000C") control code shall be used to indicate clearing of the visible text area. The End of Medium (hexadecimal "0x0019") control code shall be used to indicate "description off" status. However, this Recommendation puts no restriction on how video description text is actually displayed and stored.

W.6.3.6 Uniform Resource Identifier (URI) text

The message consists of a uniform resource identifier (URI), as defined in IETF RFC 2396. The URI shall be coded according to ISO/IEC 10646 UTF-8.

W.6.3.7 Current picture header repetition

The picture header from the current picture is repeated in this message. The repeated bits exclude any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.8 Previous picture header repetition

The picture header from the previously transmitted picture is repeated in this message. The repeated bits exclude the first two bytes of picture start code (PSC) and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.9 Next picture header repetition, reliable TR

The picture header from the next picture to be transmitted is repeated in this message. The repeated bits exclude the first two bytes of picture start code (PSC) and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.10 Next picture header repetition, unreliable TR

The picture header from the next picture to be transmitted is repeated in this message. The repeated bits exclude the first three bytes of picture header and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4. Any TR or ETR bits in the repeated picture header are not necessarily the same as the corresponding bits in the next picture header.

W.6.3.11 Interlaced field indications

In the case of interlaced field indications, the message consists of an indication of interlaced field coding. This indication does not affect the decoding process. However, it indicates that the current picture was not actually scanned as a progressive-scan picture. In other words, it indicates that the current coded picture contains only half of the lines of the full resolution source picture. DSIZE shall be 1, CONT shall be 0, and EBIT shall be 0 for interlaced field indications. In the case of interlaced field coding, each increment of the temporal reference denotes the time between the sampling of alternate half-picture fields of a picture, rather than the time between two complete

pictures. In the case of a top interlaced field indication, the current picture contains the first (i.e., top), third, fifth, etc. lines of the complete picture. In the case of a bottom interlaced field indication, the current picture contains the second, fourth, sixth, etc. lines of the complete picture. When sending interlaced field indications, an encoder shall conform to the following conventions:

1) The encoder shall use a picture clock frequency (custom picture clock frequency, if necessary) such that each new field of the original source video corresponds to an increment of 1 in the temporal reference.

2) The encoder shall use a picture size (custom picture size, if necessary) such that the picture dimensions correspond to those of a single field.

3) The encoder shall use a pixel aspect ratio (custom pixel aspect ratio, if necessary) such that the full-height picture aspect ratio corresponds to the picture aspect ratio derived from the pixel aspect ratio of the single field represented by the current encoded picture.

Interlaced field scanning was introduced originally as an analog video compression technique. Although progressive picture scanning is generally regarded as superior for digital compression and display, the use of interlaced field scanning has persisted in many camera and display designs. Interlaced field coding (which can be implemented with lower delay than either interlaced full-picture coding or progressive-scan picture coding at half the interlaced field rate) is therefore supported by the indications herein.

An encoder shall not send interlaced field indications unless the capability of the decoder to receive and properly process such field-based pictures has been established by external means (for example, ITU-T Rec. H.245). Failure to establish such a decoder capability may produce a visually annoying small-amplitude vertical shaking behaviour in the decoded picture received and displayed by a decoder.

For example, an encoder may use interlaced field coding with application of the Reference Picture Selection mode (specified in Annex N) or the Enhanced Reference Picture Selection mode (specified in Annex U) to allow the addressing of more than one prior field. For "525/60" interlaced field coding for a 4:3 picture aspect ratio with 704 coded luminance samples per line and 240 coded luminance lines per field, the encoder shall use a custom picture size having a picture width of 704 and a picture height of 240, a custom pixel aspect ratio of 5:11, and a custom picture clock frequency specified with a clock conversion code "1" and a clock divisor of 30. For "625/50" interlaced field coding for a 4:3 picture aspect ratio with 704 coded luminance samples per line and 288 coded luminance lines per field, the encoder shall use a custom picture size having a picture width of 704 and a picture height of 288, a custom pixel aspect ratio of 6:11, and a custom picture clock frequency specified with a clock conversion code "0" and a clock divisor of 36.

The vertical sampling positions of the chrominance samples in interlaced field coding of a top field picture are specified as shifted up by 1/4 luminance sample height relative to the field sampling grid in order for these samples to align vertically to the usual position relative to the full-picture sampling grid. The vertical sampling positions of the chrominance samples in interlaced field coding of a bottom field picture are specified as shifted down by 1/4 luminance sample height relative to the field sampling grid in order for these samples to align vertically to the usual position relative to the full-picture sampling grid. The horizontal sampling positions of the chrominance samples are specified as unaffected by the application of interlaced field coding. The vertical sampling positions are shown with their corresponding temporal sampling positions in Figure W.2.

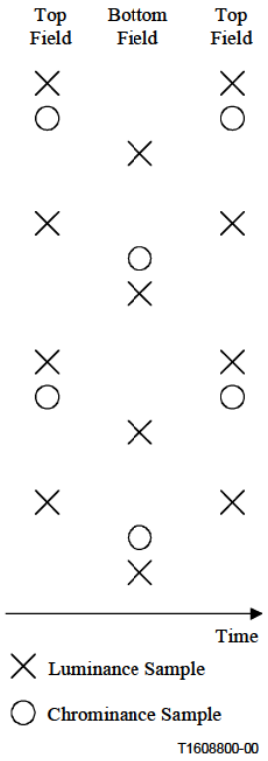


Figure W.2/H.263 – Vertical and temporal alignment of chrominance samples for interlaced field coding

W.6.3.12 Picture number

This message shall not be used if Annex U is in use. The message contains two data bytes that carry a 10-bit Picture Number. Consequently, DSIZE shall be 3, CONT shall be 0, and EBIT shall be 6. Picture Number shall be incremented by 1 for each coded and transmitted I or P picture or PB or Improved PB frame, in a 10-bit modulo operation. For EI and EP pictures, Picture Number shall be incremented for each EI or EP picture within the same scalability enhancement layer. For B pictures, Picture Number shall be incremented relative to the value in the most recent non-B picture in the reference layer of the B picture which precedes the B picture in bitstream order (a picture which is temporally subsequent to the B picture). If adjacent pictures in the same enhancement layer have the same temporal reference, and if the reference picture selection mode (see Annex N) is in use, the decoder shall regard this occurrence as an indication that redundant copies have been sent of approximately the same pictured scene content, and all of these pictures shall share the same Picture Number. If the difference (modulo 1024) of the Picture Numbers of two consecutively received non-B pictures in the same enhancement layer is not 1, and if the pictures do not represent approximately the same pictured scene content as described above, the decoder should infer a loss of pictures or corruption of data.

W.6.3.13 Spare reference pictures

Encoders can use this message to instruct decoders which pictures resemble the current motion compensation reference picture so well that one of them can be used as a spare reference picture if the actual reference picture is lost during transmission. If a decoder lacks an actual reference picture but can access a spare reference picture, it should not request for an INTRA picture update. It is up to encoders to choose the spare reference pictures, if any. The message data bytes contain the

Picture Numbers of the spare reference pictures in preference order (the most preferred appearing first). Picture Numbers refer to the values that are transmitted according to Annex U or W.6.3.12. This message can be used for P, B, PB, Improved PB, and EP picture types. However, if Annex N or Annex U is in use and if the picture is associated with multiple reference pictures, this message shall not be used. For EP pictures, the message shall be used only for forward prediction, whereas upward prediction is always done from the temporally corresponding reference layer picture. For B, PB, and Improved PB picture types, it specifies a picture for use as a forward motion prediction reference. This message shall not be used if the picture is an I or EI picture.

Annex X

Profiles and levels definition

X.1 Scope

With the variety of optional modes available in this Recommendation, it is crucial that several preferred mode combinations for operation be defined, so that option-enhanced terminals will have a high probability of connecting to each other using some syntax better than the "baseline". This annex contains a list of preferred feature combinations, which are structured into "profiles" of support. It also defines some groupings of maximum performance parameters as "levels" of support for these profiles. The primary objectives of this annex are:

1) to provide a simple means of describing or negotiating the capabilities of a decoder (by specifying profile and level parameters);
2) to encourage common enhancement features to be supported in decoders for achieving maximal interoperability; and
3) to describe feature sets chosen as particularly appropriate for addressing certain key applications.

The profiles and levels are defined in the following clauses and in Tables X.1 and X.2. The minimum picture interval as specified in Table X.2 is the minimum difference in time between the decoding of consecutive pictures in the bitstream. Support of any level other than level 45 implies support of all lower levels. Support of level 45 implies support of level 10.

X.2 Profiles of preferred mode support

The profiles of support are defined by the set of features supported in the decoder for each profile. Decoder support for a given profile implies support for all valid subset combinations of the constituent modes of that profile. This requirement exists so that the limitations placed upon an encoder's choice of mode combinations are minimized. This is in keeping with the primary objective of this annex, which is to describe which optional modes should be supported at the decoder to address key applications, rather than to enforce a particular small set of mode combinations upon the encoder.

X.2.1 The Baseline Profile (Profile 0)

The Baseline Profile, designated as Profile 0, is defined herein to provide a profile designation for the minimal "baseline" capability of this Recommendation. "Baseline" refers to the syntax of this Recommendation with no optional modes of operation. This profile of support is composed of only the baseline design.

X.2.2 H.320 Coding Efficiency Version 2 Backward-Compatibility Profile (Profile 1)

The H.320 Coding Efficiency Version 2 Backward Compatibility Profile, designated as Profile 1, is defined herein to provide compatibility with a feature set adopted into the H.242 capability exchange mechanism for use by H.320 circuit-switched terminal systems. It provides basic enhanced coding efficiency and simple enhanced functionality within the feature set available in the second version of this Recommendation (which did not include Annexes U, V and W). This profile of support is composed of the baseline design plus the following modes:

1) **Advanced INTRA Coding (Annex I)** – Use of this mode improves the coding efficiency for INTRA macroblocks (whether within INTRA pictures or predictively-coded pictures). The additional computational requirements of this mode are minimal at both the encoder and decoder (as low as a maximum of 8 additions/subtractions per 8×8 block in the decoding process plus the use of a different but very similar VLC table in order to obtain a significant improvement in coding efficiency). For these reasons, Advanced INTRA Coding is included in this basic package of support.

2) **Deblocking Filter (Annex J)** – Because of the significant subjective quality improvement that may be realized with a deblocking filter, these filters are widely in use as a method of post-processing in video communication terminals. Annex J represents the preferred mode of operation for a deblocking filter because it places the filter within the coding loop. This placement eases the implementation of the filter (by reducing the required memory) and somewhat improves the coding performance over a post-processing implementation. As with the Advanced Prediction mode, this mode also includes the four-motion-vector-per-macroblock feature and picture boundary extrapolation for motion compensation, both of which can further improve coding efficiency. The computational requirements of the deblocking filter are several hundred operations per coded macroblock, but memory accesses and computational dependencies are uncomplicated. This last point is what makes the Deblocking Filter preferable to Advanced Prediction for some implementations. Also, the benefits of Advanced Prediction are not as substantial when the Deblocking Filter is used as well. Thus, the Deblocking Filter is included in this basic package of support.

3) **Full-Picture Freeze Supplemental Enhancement Information (Annex L, clause L.4)** – The full-picture freeze is very simple to implement, requiring only that the decoder be able to stop the transfer of new pictures from its output buffer to the video display. This capability is useful for preventing the display of low-fidelity pictures while the encoder is building up a higher fidelity picture.

4) **Modified Quantization (Annex T)** – This mode includes an extended DCT coefficient range, modified DQUANT syntax, and a modified step size for chrominance. The first two features allow for more flexibility at the encoder and may actually decrease the encoder's computational load (by eliminating the need re-encode macroblocks when coefficient level saturation occurs). The third feature noticeably improves chrominance fidelity, typically with little added bit-rate cost and with virtually no increase in computation. At the decoder, the only significant added computational burden is the ability to parse some new bitstream symbols.

X.2.3 Version 1 Backward-Compatibility Profile (Profile 2)

The Version 1 Backward-Compatibility Profile, designated as Profile 2, is defined herein to provide enhanced coding efficiency performance within the feature set available in the first version of ITU-T Rec. H.263 (which did not include Supplemental Enhancement Information or any of the optional features which use PLUSPTYPE). This profile of support is composed of the baseline design plus the following single mode:

1) **Advanced Prediction (Annex F)** – From a coding efficiency standpoint, this mode is the most important of the modes available in the first version (Version 1) of this Recommendation. It includes overlapped block motion compensation, the four-motion-vector-per-macroblock feature, and it allows for motion vectors to point outside of the picture boundaries. The use of Advanced Prediction results in significant improvements in both subjective and objective performance. It does, however, require an appreciable increase in computation and introduces complicating data dependencies in the order of processing at the decoder. However, since implementations of this Recommendation that were designed prior to the adoption of the other modes in this list might have implemented Advanced Prediction by itself, Advanced Prediction-only operation is recommended for maximal quality with backward compatibility to version 1 decoders.

X.2.4 Version 2 Interactive and Streaming Wireless Profile (Profile 3)

The Version 2 Interactive and Streaming Wireless Profile, designated as Profile 3, is defined herein to provide enhanced coding efficiency performance and enhanced error resilience for delivery to wireless devices within the feature set available in the second version of this Recommendation (which did not include Annexes U, V, and W). This profile of support is composed of the baseline design plus the following modes:

1) **Advanced INTRA Coding (Annex I)** – See X.2.2 item 1.

2) **Deblocking Filter (Annex J)** – See X.2.2 item 2.

3) **Slice Structured Mode (Annex K)** – The Slice Structured mode is included here due to its enhanced ability to provide resynchronization points within the video bitstream for recovery from erroneous or lost data. Support for the Arbitrary Slice Ordering (ASO) and Rectangular Slice (RS) submodes of the Slice Structured mode are not included in this profile, in order to limit the complexity requirements of the decoder. The additional computational burden imposed by the Slice Structured mode is minimal, limited primarily to bitstream generation and parsing.

4) **Modified Quantization (Annex T)** – See X.2.2 item 4.

X.2.5 Version 3 Interactive and Streaming Wireless Profile (Profile 4)

The Version 3 Interactive and Streaming Wireless Profile, designated as Profile 4, is defined herein to provide enhanced coding efficiency performance and enhanced error resilience for delivery to wireless devices, while taking advantage of the enhanced features of the third version of this Recommendation. This profile of support is composed of the baseline design, plus the following additional features as follows:

1) **Profile 3** – This feature set provides several enhancements useful for support of wireless video transmission.

2) **Data Partitioned Slice Mode (Annex V)** – This feature enhances error resilience performance by separating motion vector data from DCT coefficient data within slices, and protects the motion vector information (the most important part of the detailed macroblock data) by using reversible variable-length coding. Support of the Arbitrary Slice Ordering (ASO) and Rectangular Slice (RS) submodes are not included in this profile, in order to limit the complexity requirements of the decoder.

3) **Previous Picture Header Repetition Supplemental Enhancement Information (Annex W, clause W.6.3.8)** – This feature allows the decoder to receive and recover the header information from a previous picture in case of data loss or corruption.

X.2.6 Conversational High Compression Profile (Profile 5)

The Conversational High Compression Profile, designated as Profile 5, is defined herein to provide enhanced coding efficiency performance without adding the delay associated with the use of B pictures and without adding error resilience features. This profile of support is composed of the baseline design, plus the following additional features as follows:

1) **Profile 1** – This feature set provides several enhancements useful for enhanced coding efficiency.

2) **Profile 2** – This profile adds the Advanced Prediction mode (Annex F), which provides a further enhancement of coding efficiency performance and backward-compatibility with implementations of the first version of this Recommendation.

3) **Unrestricted Motion Vectors with UUI = "1" (Annex D)** – Annex D has two primary features:

   a) picture boundary extrapolation; and

   b) longer motion vector support.

   The first of these features is already supported by the inclusion of Annex J in Profile 1. The longer motion vector support can provide a significant improvement in coding efficiency, especially for large picture sizes, rapid motion, camera movement, and low picture rates. When used with PLUSPTYPE present, this mode also allows for longer motion vector differences, which can significantly simplify encoder operation. The longer motion vectors do present a potential problem for the decoder in terms of memory access, but picture-size-dependent limits on the maximum motion vector size prevent this problem from becoming an appreciable obstacle to implementation.

4) **Enhanced Reference Picture Selection (Annex U)** – This mode adds a significant gain in compression efficiency performance due to the ability to use multiple prior pictures as reference data for macroblock-level prediction of the subsequent pictures. The Sub-Picture Removal submode (Annex U, clause U.4.3) of the Enhanced Reference Picture Selection mode is not included in Profile 5.

X.2.7 Conversational Internet Profile (Profile 6)

The Conversational Internet Profile, designated as Profile 6, is defined herein to provide enhanced coding efficiency performance without adding the delay associated with the use of B pictures, but adding some error resilience suitable for use on Internet Protocol (IP) networks (which use packet-based data protocols with relatively large packets and which exhibit data losses rather than data corruption). This profile of support is composed of the baseline design, plus the following additional features as follows:

1) **Profile 5** – This feature set provides several enhancements useful for enhanced coding efficiency.

2) **Slice Structured mode (Annex K) with Arbitrary Slice Ordering (ASO) submode** – The Slice Structured mode is included here due to its enhanced ability to provide resynchronization points within the video bitstream for recovery from lost data packets. The Arbitrary Slice Ordering (ASO) submode of the Slice Structured mode is also included in order to allow for interleaved packetization for motion-compensated error concealment and for out-of-sequence data reception. Support for the Rectangular Slice (RS) submode of the Slice Structured mode is not included in this profile, in order to limit the complexity requirements of the decoder. The additional computational burden imposed by the Slice Structured mode is minimal, limited primarily to bitstream generation and parsing.

X.2.8 Conversational Interlace Profile (Profile 7)

The Conversational Interlace Profile, designated as Profile 7, is defined herein to provide enhanced coding efficiency performance for low-delay applications, plus support of interlaced video sources. This profile of support is composed of the baseline design, plus the following additional features as follows:

1) **Profile 5** – This feature set provides several enhancements useful for enhancing coding efficiency without adding delay.

2) **Interlaced Field Indications For 240-line and 288-line Pictures (Annex W, clause W.6.3.11)** – This feature allows video to be sent in an interlaced source picture format for compatibility with existing camera designs.

X.2.9 High Latency Profile (Profile 8)

The High Latency Profile, designated as Profile 8, is defined herein to provide enhanced coding efficiency performance for applications without critical delay constraints. This profile of support is composed of the baseline design, plus the following additional features as follows:

1) **Profile 6** – This feature set provides several enhancements useful for enhanced coding efficiency and robustness to data losses.

2) **Reference Picture Resampling (Implicit Factor-of-4 Mode Only) (Annex P, clause P.5)** – The implicit factor-of-4 mode of Reference Picture Resampling allows for automatic reference picture resampling only when the size of the new frame is changed, as indicated in the picture header. No bitstream overhead is required for this mode of operation. Predictive dynamic resolution changes allow an encoder to make intelligent trade-offs between temporal and spatial resolution. Furthermore, this simplest mode of operation for Annex P (factor-of-4 upsampling or downsampling only) adds only a modest amount of computational complexity to both the encoder or decoder, since the factor-of-4 case uses a simple fixed FIR filter (requiring roughly 4 operations per pixel, at most).

3) **B Pictures (Temporal Scalability, Annex O, clause O.1.1)** – This feature consists of B pictures, which are pictures allowing bidirectional temporal prediction. The addition of B pictures enhances coding efficiency performance, but at some cost in added processing power and encoding and decoding delay. The two-picture backward prediction submode for B pictures in Enhanced Reference Picture Selection mode (Annex U, clause U.3.1.5.5) is not supported in Profile 8.

X.3 Picture formats and picture clock frequencies

To ensure a high quality level of interoperability, encoders and decoders supporting a large standard picture format (QCIF, CIF, 4CIF, 16CIF) should support all smaller standard picture formats. This is a requirement of all decoders conforming to the profiles and levels defined in this annex. (As specified elsewhere in this Recommendation, decoders shall support sub-QCIF and QCIF, and encoders shall support sub-QCIF or QCIF.) For example, a decoder conforming to a profile and level defined in this annex which is capable of decoding 4CIF pictures shall also support the decoding of CIF pictures.

Decoders should be capable of operation with a smaller picture format at maximum picture rates no lower than the maximum picture rate for which it is capable of operation with a larger standard picture format. This is a requirement of all decoders conforming to the profiles and levels defined in this annex. For example, a decoder conforming to a profile and level defined in this annex which is capable of decoding 4CIF pictures at 25 pictures per second shall also be able to decode CIF, QCIF and SQCIF pictures at least at 25 pictures per second.

Encoders and decoders supporting custom picture formats and/or custom picture clock frequencies are recommended to follow the rules defined in this paragraph. These rules are requirements of all decoders conforming to the profiles and levels defined in this annex:

1) A decoder for any profile and level defined herein that supports a maximum picture format shall support all standard picture formats smaller or equal in both height and width than those of the maximum supported picture format. For example, a decoder supporting a custom picture format of 720 × 288 shall support CIF, QCIF and sub-QCIF picture decoding.

2) A decoder for any profile and level defined herein that supports custom picture formats shall support all standard or custom picture formats having both height and width smaller than or equal to those of the maximum supported picture format.

3) A decoder for any profile and level defined herein that supports a minimum picture interval with the standard picture clock frequency of (30 000)/1001 units per second shall support the same or smaller minimum picture interval for all supported picture formats having both height and width smaller than or equal to those of the maximum picture format at which the minimum picture interval is specified.

4) A decoder for any profile and level defined herein that supports a minimum picture interval and supports custom picture clock frequencies shall support the use of any picture clock frequency with the same or larger picture interval for all supported picture formats having both height and width smaller than or equal to those of the maximum picture format at which the minimum picture interval is specified.

X.4 Levels of performance capability

Eight levels of performance capability are defined for decoder implementation. The Hypothetical Reference Decoder has the minimal size specified in Table X.1 for all levels of Profiles 0 through 4. In Profiles 5 though 8 the Hypothetical Reference Decoder has an increased size and Enhanced Reference Picture Selection is supported with multiple reference pictures. Table X.2 defines the detailed performance parameters of each of these levels:

1) **Level 10** – Support of QCIF and sub-QCIF resolution decoding, capable of operation with a bit rate up to 64 000 bits per second with a picture decoding rate up to (15 000)/1001 pictures per second.

2) **Level 20** – Support of CIF, QCIF and sub-QCIF resolution decoding, capable of operation with a bit rate up to 2·(64 000) = 128 000 bits per second with a picture decoding rate up to (15 000)/1001 pictures per second for CIF pictures and (30 000)/1001 pictures per second for QCIF and sub-QCIF pictures.

3) **Level 30** – Support of CIF, QCIF and sub-QCIF resolution decoding, capable of operation with a bit rate up to 6·(64 000) = 384 000 bits per second with a picture decoding rate up to (30 000)/1001 pictures per second.

4) **Level 40** – Support of CIF, QCIF and sub-QCIF resolution decoding, capable of operation with a bit rate up to 32·(64 000) = 2 048 000 bits per second with a picture decoding rate up to (30 000)/1001 pictures per second.

4.5) **Level 45** – Support of QCIF and sub-QCIF resolution decoding, capable of operation with a bit rate up to 2·(64 000) = 128 000 bits per second with a picture decoding rate up to (15 000)/1001 pictures per second. Additionally, in profiles other than profiles 0 and 2, support of custom picture formats of size QCIF and smaller.

5) **Level 50** – Support of custom and standard picture formats of size CIF and smaller, capable of operation with a bit rate up to 64·(64 000) = 4 096 000 bits per second with a picture decoding rate up to 50 pictures per second for CIF or smaller picture formats and up to (60 000)/1001 pictures per second for 352 × 240 and smaller picture formats.

6) **Level 60** – Support of custom and standard picture formats of size 720 × 288 and smaller, capable of operation with a bit rate up to 128·(64 000) = 8 192 000 bits per second with a picture decoding rate up to 50 pictures per second for 720 × 288 or smaller picture formats and up to (60 000)/1001 pictures per second for 720 × 240 and smaller picture formats.

7) **Level 70** – Support of custom and standard picture formats of size 720 × 576 and smaller, capable of operation with a bit rate up to 256·(64 000) = 16 384 000 bits per second with a picture decoding rate up to 50 pictures per second for 720 × 576 or smaller picture formats and up to (60 000)/1001 pictures per second for 720 × 480 and smaller picture formats.

The bit rate at which a particular profile and level are used in a system shall never exceed that specified in this annex. However, particular systems may include other means to signal further limits on the bit rate. Other aspects of profile and level capabilities may also be subject to additional capability restrictions when used in particular systems, but the capabilities required for decoding any bitstream for a particular profile and level defined herein shall never exceed those specified in this annex.

Table X.1/H.263 – Summary of profiles

Annex/clause below for profile listed at the right	0	1	2	3	4	5	6	7	8
5.1.5: *Custom Picture Format (CPFMT)*	L	L	L	L	L	L	L	L	L
5.1.7: *Custom Picture Clock Frequency Code (CPCFC)*	L	L	L	L	L	L	L	L	L
C: *Continuous Presence Multipoint and Video Mux*									
D.1: *Motion vectors over picture boundaries*		X	X	X	X	X	X	X	X
D.2 with UUI = '1' or UUI not present: *Extension of the motion vector range*						X	X	X	X
D.2 with UUI = '01': *Unlimited extension of the motion vector range*									
E: *Syntax-based Arithmetic Coding*									
F.2: *Four motion vectors per macroblock*		X	X	X	X	X	X	X	X
F.3: *Overlapped block motion compensation*			X			X	X	X	X
G: *PB-Frames*									
H: *Forward Error Correction (use may be imposed at system level as in ITU-T Rec. H.320)*									
I: *Advanced Intra Coding*		X		X	X	X	X	X	X
J: *Deblocking Filter*		X		X	X	X	X	X	X
K without submodes: *Slice Structured Coding – Without submodes*				X	X		X		X
K with ASO. *Slice Structured Coding – With Arbitrary Slice Ordering submode*							X		X
K with RS: *Slice Structured Coding – With Rectangular Slice submode*									
L.4: *Supplemental Enhancement Full picture freeze*		X				X	X	X	X
L: *Supplemental Enhancement – Other SEI features*									
M: *Improved PB-Frames*									
N: *Reference Picture Selection (and submodes)*									
O.1.1 Temporal (B pictures): *Temporal, SNR, and Spatial Scalability – B pictures for Temporal Scalability*									X

Table X.1/H.263 – Summary of profiles

Annex/clause below for profile listed at the right	0	1	2	3	4	5	6	7	8
O SNR and Spatial: *Temporal, SNR, and Spatial Scalability – EI and EP pictures for SNR and Spatial Scalability*									
P.5: *Reference Picture Resampling – Implicit Factor of Four*									X
P: *Reference Picture Resampling – More General Resampling*									
Q: *Reduced Resolution Update*									
R: *Independent Segment Decoding*									
S: *Alternative Inter VLC*									
T: *Modified Quantization*		X		X	X	X	X	X	X
U without submodes: *Enhanced Reference Picture Selection – Without submodes*						X	X	X	X
U with SPR: *Enhanced Reference Picture Selection – With Sub-Picture Removal submode*									
U with BTPSM: *Enhanced Reference Picture Selection – With B-Picture Two-Picture submode*									
V: *Data Partitioned Slices*					X				
W.6.3.8: *Additional SEI Specification – Prior Picture Header Repetition*					X				
W.6.3.11: *Additional SEI Specification – Interlaced Field Indications*								X	
W: *Additional SEI Specification – Other SEI features*									
"X" indicates that support of a feature is part of a profile. "L" indicates that the inclusion of a feature depends on the level within the profile.									

Table X.2/H.263 – Levels of operation

Parameter below for level listed at the right	10	20	30	40	45	50	60	70
Max picture format	QCIF (176 × 144)	CIF (352 × 288)	CIF (352 × 288)	CIF (352 × 288)	QCIF (176 × 144) support of CPFMT in profiles other than 0 and 2	CIF (352 × 288) support of CPFMT	CPFMT: 720 × 288 support of CPFMT	CPFMT: 720 × 576 support of CPFMT
Min picture interval	2002/(30 000) s	2002/(30 000) s for CIF 1001/(30 000) s for QCIF and sub-QCIF	1001/(30 000) s	1001/(30 000) s	2002/(30 000) s support of CPCFC in profiles other than 0 and 2	1/50 s at CIF or lower 1001/(60 000) s at 352 × 240 or smaller support of CPCFC	1/50 s at 720 × 288 or lower 1001/(60 000) s at 720 × 240 or smaller support of CPCFC	1/50 s at 720 × 576 or lower 1001/(60 000) s at 720 × 480 or smaller support of CPCFC
Max bit rate in 64 000 bits/s units	1	2	6	32	2	64	128	256
Max HRD B in 16 384 bit units	1: Prof. 5-8	2: Prof. 5-8	6: Prof. 5-8	32: Prof. 5-8	2: Prof. 5-8	64: Prof. 5-8	64: Prof. 5-8	256: Prof. 5-8
Max BPPmaxKb in 1024 bit units	128: Prof. 5-8	512: Prof. 5-8	512: Prof. 5-8	512: Prof. 5-8	128: Prof. 5-8	512: Prof. 5-8	1024: Prof. 5-8	1024: Prof. 5-8
Max ERPS reference pictures (Annex U)	5: Prof. 5-7 10: Prof. 8	5: Prof. 5-7 10: Prof. 8 multiplied by 2 for QCIF or sub-QCIF in Prof. 5-8	5: Prof. 5-7 10: Prof. 8 multiplied by 2 for QCIF or sub-QCIF in Prof. 5-8	5: Prof. 5-7 10: Prof. 8 multiplied by 2 for QCIF or sub-QCIF in Prof. 5-8	5: Prof. 5-7 10: Prof. 8	5: Prof. 5-7 10: Prof. 8 multiplied by 2 for QCIF or smaller in Prof. 5-8	5: Prof. 5-7 10: Prof. 8 multiplied by 2 for CIF or smaller, and by 4 for QCIF or smaller in Prof. 5-8	5: Prof. 5-7 10: Prof. 8 multiplied by 2 for CIF or smaller, and by 4 for QCIF or smaller in Prof. 5-8

NOTE 1 – In profiles for which a maximum number of reference picture buffers is not specified in Table X.2, no support for multiple reference picture buffering is required.

NOTE 2 – In profiles for which a maximum BPPmaxKb and HRD B are not specified in Table X.2, the minimum value specified in Table X.1 applies for the specified maximum bit rate and resolution.

X.5 Generic capability definitions for use with ITU-T Rec. H.245

Table X.3 defines a capability identifier for establishing H.263 capabilities for use in systems that use ITU-T Rec. H.245 for capability determination. These parameters shall only be included as **genericVideoCapability** within the **VideoCapability** structure and as **genericVideoMode** within the **VideoMode** structure of ITU-T Rec. H.245. Tables X.4 to X.14 define the associated capability parameters.

When included in Logical Channel Signalling or Mode Request, exactly one parameter with Parameter identifier value in the range zero to eight shall be present; that is, only one profile shall be specified.

Table X.3/H.263 – Capability identifier for H.263 capability

Capability name:	H.263
Capability class:	Video codec
Capability identifier type:	Standard
Capability identifier value:	itu-t (0) recommendation (0) h (8) 263 generic-capabilities (1) 0
MaxBitRate:	The maxBitRate field shall always be included.
NonCollapsingRaw:	This field shall not be included.
Transport:	This field shall not be included.

Table X.4/H.263 – The Baseline Profile (Profile 0) capability

Parameter name:	baselineProfile
Parameter description:	This is a collapsing GenericParameter. baselineProfile indicates the maximum level of support for the Baseline Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	0
Parameter status:	Mandatory
Parameter type:	unsignedMin
Supersedes:	–

Table X.5/H.263 – H.320 Coding Efficiency Version 2 Backward-Compatibility Profile (Profile 1) capability

Parameter name:	h320Profile
Parameter description:	This is a collapsing GenericParameter. h320Profile indicates the maximum level of support for the H.320 Coding Efficiency Version 2 Backward-Compatibility Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	1
Parameter status:	Optional
Parameter type:	unsignedMin
Supersedes:	–

Table X.6/H.263 – Version 1 Backward-Compatibility Profile (Profile 2) capability

Parameter name:	backwardCompatibleProfile
Parameter description:	This is a collapsing GenericParameter. backwardCompatibleProfile indicates the maximum level of support for the Version 1 Backward-Compatibility Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	2
Parameter status:	Optional
Parameter type:	unsignedMin
Supersedes:	–

Table X.7/H.263 – Version 2 Interactive and Streaming Wireless Profile (Profile 3) capability

Parameter name:	v2WirelessProfile
Parameter description:	This is a collapsing GenericParameter. v2WirelessProfile indicates the maximum level of support for the Version 2 Interactive and Streaming Wireless Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	3
Parameter status:	Optional
Parameter type:	unsignedMin
Supersedes:	–

Table X.8/H.263 – Version 3 Interactive and Streaming Wireless Profile (Profile 4) capability

Parameter name:	v3WirelessProfile
Parameter description:	This is a collapsing GenericParameter. v3WirelessProfile indicates the maximum level of support for the Version 3 Interactive and Streaming Wireless Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	4
Parameter status:	Optional
Parameter type:	unsignedMin
Supersedes:	–

Table X.9/H.263 – Conversational High Compression Profile (Profile 5) capability

Parameter name:	conversationalProfile
Parameter description:	This is a collapsing GenericParameter. conversationalProfile indicates the maximum level of support for the Conversational High Compression Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	5
Parameter status:	Optional
Parameter type:	unsignedMin
Supersedes:	–

Table X.10/H.263 – Conversational Internet Profile (Profile 6) capability

Parameter name:	conversationalInternetProfile
Parameter description:	This is a collapsing GenericParameter. conversationalInternetProfile indicates the maximum level of support for the Conversational Internet Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	6
Parameter status:	Optional
Parameter type:	unsignedMin
Supersedes:	–

Table X.11/H.263 – Conversational Plus Interlace Profile (Profile 7) capability

Parameter name:	conversationalInterlaceProfile
Parameter description:	This is a collapsing GenericParameter. conversationalInterlaceProfile indicates the maximum level of support for the Conversational Plus Interlace Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	7
Parameter status:	Optional
Parameter type:	unsignedMin
Supersedes:	–

Table X.12/H.263 – High Latency Profile (Profile 8) capability

Parameter name:	highLatencyProfile
Parameter description:	This is a collapsing GenericParameter. highLatencyProfile indicates the maximum level of support for the High Latency Profile when present in Capability Exchange, the maximum level to be transmitted when present in Logical Channel Signalling, and the desired level when present in Mode Request.
Parameter identifier value:	8
Parameter status:	Optional
Parameter type:	unsignedMin
Supersedes:	–

Table X.13/H.263 – Temporal Spatial Trade-Off capability

Parameter name:	temporalSpatialTradeOffCapability
Parameter description:	This is a collapsing GenericParameter. The presence of this parameter indicates that the encoder is able to vary its trade-off between temporal and spatial resolution as commanded by the remote terminal. It has no meaning when part of a receive capability.
Parameter identifier value:	9
Parameter status:	Optional
Parameter type:	logical
Supersedes:	–

Table X.14/H.263 – Video Bad Macroblocks capability

Parameter name:	videoBadMBsCap
Parameter description:	This is a collapsing GenericParameter. The presence of this parameter indicates the capability of an encoder to receive or a decoder to transmit the videoBadMBs command. When part of a transmit capability, it indicates the ability of the encoder to process videoBadMBs commands and to take appropriate corrective action toward recovery of video quality. When part of a receive capability, it indicates the ability of the decoder to send appropriate videoBadMBs indications.
Parameter identifier value:	10
Parameter status:	Optional
Parameter type:	logical
Supersedes:	–

Appendix I

Error tracking

I.1 Introduction

This appendix describes a method to recover efficiently after transmission errors if erroneous MBs are reported via a feedback channel to the encoder. The capability of sending and processing feedback information is signalled via external means (for example, by ITU-T Rec. H.245). Furthermore, format and content of the feedback message are defined externally (for example, by ITU-T Rec. H.245).

I.2 Error tracking

Because INTRA coding stops temporal error propagation, it should be used for macroblocks which are severely affected by transmission errors. This requires that the location and extent of image artefacts can be made available to the encoder. The following algorithm provides an estimated error distribution based on feedback information received by the encoder. It considers spatial error propagation caused by motion-compensated prediction as well as the delay until the reception of the feedback message. The algorithm illustrates one possible approach to evaluate feedback messages for spatio-temporal error tracking. Other algorithms are possible.

Assume N macroblocks within each frame enumerated mb = 1...N from top-left to bottom-right. Let $\{n_{err}, mb_{first}, mb_{last}\}$ be the feedback message to the encoder, where $mb_{first} \leq mb \leq mb_{last}$ indicates a set of erroneous macroblocks in frame n_{err}.

To evaluate the feedback message, the encoder must continuously record information during the encoding of each frame. First, the initial error $E_0(mb, n)$ that would be introduced by the loss of macroblock mb in frame n needs to be stored. Assuming a simple error concealment where erroneous macroblocks are treated as not coded, $E_0(mb, n)$ is computed as the Summed Absolute Difference (SAD) of macroblock mb in frame n and n – 1. Second, the number of pixels transferred from macroblock mb_{source} in frame n – 1 to macroblock mb_{dest} in frame n is stored in dependencies $d(mb_{source}, mb_{dest}, n)$. These dependencies are derived from the motion vectors.

Assume that a feedback message arrives before frame n_{next} is encoded, such that $n_{next} > n_{err}$. Then, the estimated error E(mb, n_{err}) in macroblock mb and frame n_{err} is initialized as:

$$E(mb, n_{err}) = \begin{cases} E_0(mb, n_{err}) & \text{for } mb_{first} \le mb \le mb_{last} \\ 0 & \text{else} \end{cases}$$

For subsequent frames n, with $n_{err} < n < n_{next}$, the error may be estimated as:

$$E(mb, n) = \sum_{i=1}^{N} E(i, n-1) \frac{d(i, mb, n)}{256}$$

where a uniformly distributed error in each macroblock is assumed after each iteration.

The estimated error E(mb, $n_{next} - 1$) is incorporated into the mode decision of the next frame. For example, macroblock mb is coded in INTRA mode, if E(mb, $n_{next} - 1$) exceeds a threshold.

In practice, error tracking information will only be stored for the latest M frames. Then, if $n_{err} < n_{next} - M$, no error-tracking information is available and the encoder must take special action. For example, the next frame may be coded in INTRA mode. However, other procedures are possible and may be more effective.

Appendix II

Recommended optional enhancement

The content of the H.263 Appendix II approved in February 1998 has become obsolete after the approval of H.263 Annex X. Since this appendix is used as a reference by many users of ITU-T texts, this appendix is provided mainly as a reference for users unaware of the contents of H.263 Annex X.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks, open system communications and security
Series Y	Global information infrastructure, Internet protocol aspects and next-generation networks
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2005