Telecommunication Standardization Sector
Study Group 15
Experts Group for ATM Video Coding
(Rapporteur's Group on Part of Q.2/15)

Document AVC-597
October 1993

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**
**ORGANISATION INTERNATIONALE DE NORMALISATION**
**ISO-IEC/JTC1/SC29/WG11**
**CODED REPRESENTATION OF PICTURE AND AUDIO INFORMATION**

ISO-IEC/JTC1/SC29/WG11/MPEG93/___

**Source:** M.E.Nilsson, D.G.Morrison, BT Labs
**Title:** Proposal for modified syntax, semantics and decoding process for spatial scalability
**Purpose:** Proposal

*This document describes the syntax, semantics and decoding process used for successful bitstream exchange for spatial scalability. It is proposed that this text replace the corresponding text in the Working Draft.*

Section 6.2.2

| sequence_scalable_extension() { | No. of bits | Mnemonic |
|---|---|---|
| **extension_start_code** | 32 | bslbf |
| **extension_start_code_identifier** | 4 | uimsbf |
| **scalable_mode** | 2 | uimsbf |
| **layer_id** | 4 | uimsbf |
| if (scalable_mode = = spatial_scalable) { | | |
| **lower_layer_prediction_horizontal_size** | 14 | uimsbf |
| **marker_bit** | 1 | "1" |
| **lower_layer_prediction_vertical_size** | 14 | uimsbf |
| **horizontal_subsampling_factor_m** | 5 | uimsbf |
| **horizontal_subsampling_factor_n** | 5 | uimsbf |
| **vertical_subsampling_factor_m** | 5 | uimsbf |
| **vertical_subsampling_factor_n** | 5 | uimsbf |
| } | | |
| | | |
| if ( scalable_mode= = "temporal scalability" ) | | |
| **picture_mux_enable** | 1 | uimsbf |
| **picture_mux_order** | 3 | uimsbf |
| **picture_mux_factor** | 4 | uimsbf |
| } | | |
| next_start_code() | | |
| } | | |

### 6.3.7 Sequence scalable extension

**lower_layer_prediction_horizontal_size** -- this a 14 bit unsigned integer indicating the horizontal size of the lower layer picture which is used for spatial prediction.

**lower_layer_prediction_vertical_size** -- this a 14 bit unsigned integer indicating the vertical size of the lower layer picture which is used for spatial prediction.
**scalable_mode** - The scalable_mode indicates the type of scalability used for the picture. If no picture extension data is present for a picture, no scalability is used for that picture. It also indictates the macroblock_type tables to be used. If no picture extension data is present for a picture, tables B.2-1, B.2-2 and B.2-3 should be used.

**Table 6-?. Definition of scalable_mode**

| codeword | scalable_mode | macroblock_type tables |
|---|---|---|
| 00 | data partitioning | B.2-1, B.2-2 and B.2-3 |
| 01 | spatial scalability | B.2-5, B.2-6 and B.2-7 |
| 10 | snr scalability | ? |
| 11 | temporal scalability | B.2-1, B.2-2 and B.2-3 |

**horizontal_subsampling_factor_m** — This affects the spatial scalable upsampling process, as defined in section 7.7. The value should be non-zero.
**horizontal_subsampling_factor_n** — This affects the spatial scalable upsampling process, as defined in section 7.7. The value should be non-zero.
**vertical_subsampling_factor** -- This affects the spatial scalable upsampling process, as defined in section 7.7. The value should be non-zero.
**vertical_subsampling_factor_n** — This affects the spatial scalable upsampling process, as defined in section 7.7. The value should be non-zero.

Section 6.2.4

| picture_spatial_scalable_extension() { | No. of bits | Mnemonic |
|---|---|---|
| extension_start_code | 32 | bslbf |
| extension_start_code_identifier | 4 | uimsbf |
| lower_layer_temporal_reference | 10 | uimsbf |
| lower_layer_horizontal_offset | 15 | simsbf |
| marker_bit | 1 | "1" |
| lower_layer_vertical_offset | 15 | simsbf |
| spatial_temporal_weight_code_table_index | 2 | uimsbf |
| lower_layer_progressive_frame | 1 | uimsbf |
| lower_layer_deinterlaced_field_select | 1 | uimsbf |
| next_start_code() | | |
| } | | |

### 6.3.12 Picture spatial scalable extension

**lower_layer_temporal_reference** - An unsigned integer value which indicates the temporal reference of the lower layer picture to be used to provide the prediction. If the lower layer indicates the temporal reference with more than 10 bits, the least significant bits are indicated here. If the lower layer indicates the temporal reference with less than 10 bits, all bits are indicated here, with the extra most significant bits being set to zero.
**lower_layer_horizontal_offset** - This affects the spatial scalable upsampling process, as in section 7.7.
**lower_layer_vertical_offset** - This affects the spatial scalable upsampling process, as in section 7.7.
**spatial_temporal_weight_code_table_index** -- this indicates which table of spatial temporal weight codes is to be used as defined in section 7.7. Spatial_temporal_weight_code_table_index must be "00" when picture_structure is top_field or bottom_field and when progressive_frame is 0.

**lower_layer_progressive_frame** -- this flag is set to 0 if the lower layer picture is an interlaced frame and to "1" if the lower layer picture is a progressive frame. The use of this flag in the spatial scalable upsampling process is defined in section 7.7.
**lower_layer_deinterlaced_field_select** -- This affects the spatial scalable upsampling process, as defined in section 7.7.

## 7.7 Spatial Scalability

This clause specifies the additional decoding process required for the spatial scalable extensions. Figure 7.? is a diagram of the video decoding process with spatial scalability. The diagram is simplified for clarity.

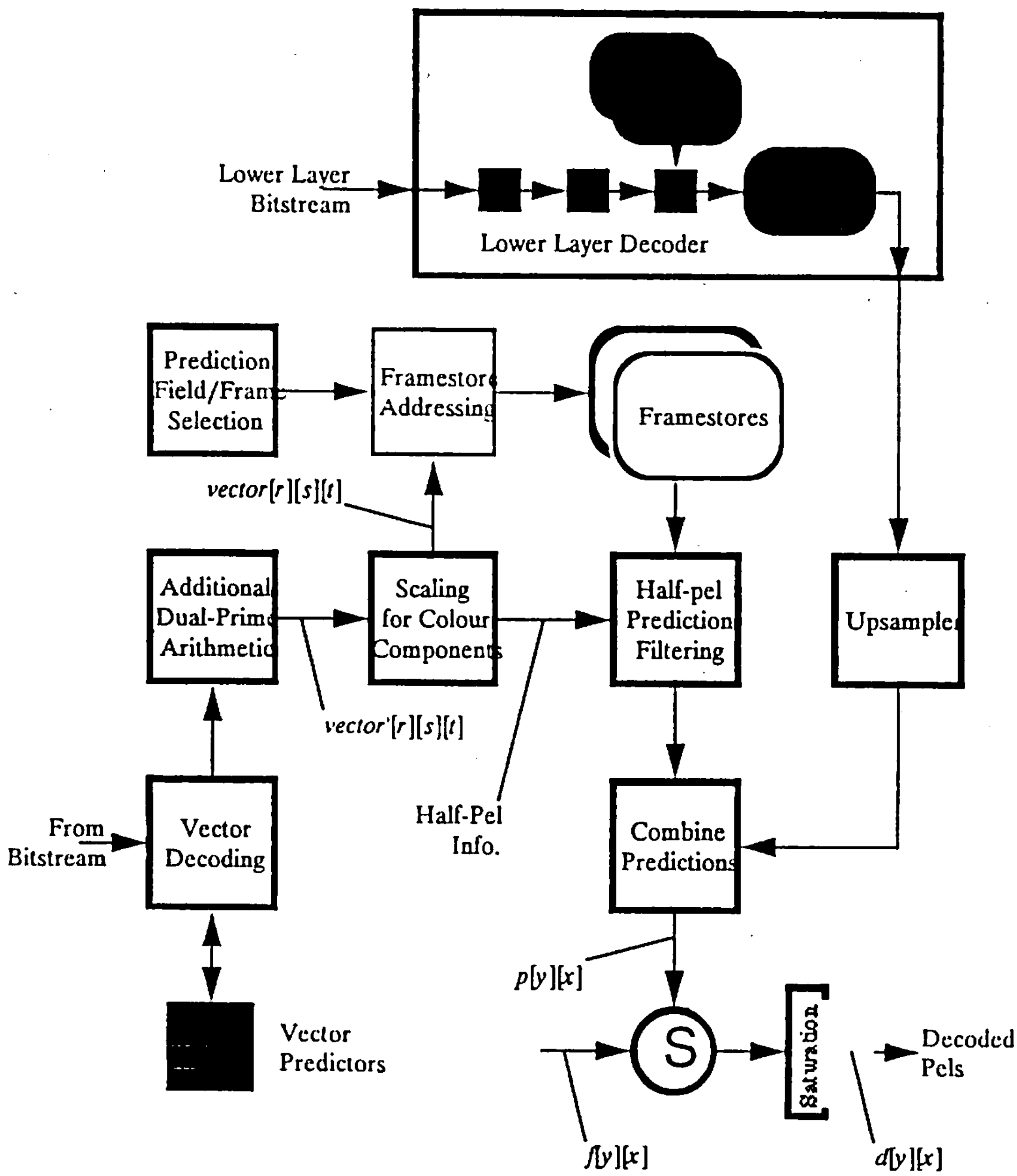


**Figure 7-?. Simplified motion compensation process for spatial scalability**

### 7.7.1 Prediction in scalable layer

A motion compensated 'temporal' prediction is made from previously decoded pictures in the current layer as described in section 7.6. In addition, a 'spatial' prediction is formed, which is an upsampled

version of the lower layer decoded picture, as described in section 7.7.2. These predictions are selected individually or combined to form the actual prediction.
In general up to four separate predictions are formed for each block which are combined together to form the final prediction block p[y][x].
In the case where a block is not coded, either because the entire macroblock is skipped or the specific block is not coded as there is no coefficient data. In this case f[y][x] is zero and the decoded pels are simply the prediction, p[y][x].

### 7.7.2 Formation of 'spatial' prediction

The spatial prediction is made from the decoded picture of the lower layer referenced by the lower_layer_temporal_reference. This prediction is made by scaling the lower layer to the same pel grid as the current layer. A 16x16 region is chosen, which corresponds to the position of the current macroblock which is being decoded.
This interpolation process is described in this section illustrated in figure 7-(?+1).

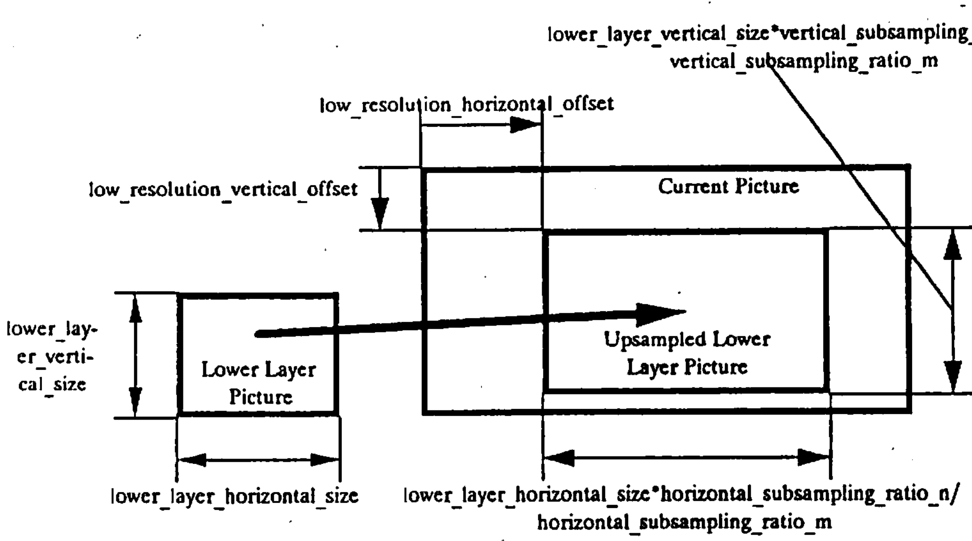


**Figure 7-?. Formation of the 'spatial' prediction by interpolation of the lower layer picture**

#### 7.7.2.1 General

The process for resampling depends on whether the lower layer is interlaced or progressive, as indicated by lower_layer_progressive_frame and whether the current layer is interlaced or progressive, as indicated by progressive_frame.

When lower_layer_progressive_frame is "1", the lower layer picture is resampled vertically as described in section 7.7.2.3. The resulting frame is considered to be progressive if progressive_frame is "1" and interlaced if progressive_frame is "0". The resulting frame is resampled horizontally as described in section 7.7.2.4. Lower_layer_deinterlaced_field_select should have the value "1".

When lower_layer_progressive_frame is "0" and progressive_frame is "0", the lower layer picture is deinterlaced as described in section 7.7.2.2, to produce two progressive fields. Both of these fields are resampled vertically as described in section 7.7.2.3. The resulting fields are subsampled to produce two interlaced fields: the even lines are taken from the first to make the top field and the odd lines are taken from the second to make the bottom field (counting lines from 0). The resulting frame is resampled horizontally as described in section 7.7.2.4. Lower_layer_deinterlaced_field_select should have the value "1".

When lower_layer_progressive_frame is "0" and progressive_frame is "1", the lower layer picture is deinterlaced as described in section 7.7.2.2, to produce two progressive fields. Only one of these fields is required. When lower_layer_deinterlaced_field_select is "0" the first is used, otherwise the second is used. The one that is used is resampled vertically as described in section 7.7.2.3. The resulting frame is resampled horizontally as described in section 7.7.2.4.

In all three cases, the resulting frame is offset by low_resolution_horizontal_offset and low_resolution_vertical_offset to produce the spatial prediction.

The upsampling process is summarised in the table below.

| Lower_layer_ field_select | Lower_layer_ progressive_frame | Progressive_ frame | spatial_temporal_ weight_code_ table_index | Apply deinterlace process | Entity used for prediction |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 00 | yes | top field |
| 1 | 0 | 1 | 00 | yes | bottom field |
| 1 | 1 | 1 | 00 | no | frame |
| 1 | 1 | 0 | 00,01,10,11 | no | frame |
| 1 | 0 | 0 | 00,01,10,11 | yes | both fields |

#### 7.7.2.2 Deinterlacing

First, the lower layer frame is padded with zeros to form a progressive grid at a picture rate equal to the field rate of the lower layer, and with the same number of lines and pels per line as the lower layer frame. It is then filtered using the relevant two field aperture filter from the table below. The temporal and vertical columns of the table indicate the relative spatial and temporal coordinates of the pels to which the filter taps defined in the other two columns apply. An intermediate sum is formed by adding the multiplied coefficients together.

| Temporal | Vertical | Filter for first field | Filter for second field |
|---|---|---|---|
| -1 | -2 | 0 | -1 |
| -1 | 0 | 0 | 2 |
| -1 | 2 | 0 | -1 |
| 0 | -1 | 8 | 8 |
| 0 | 0 | 16 | 16 |
| 0 | 1 | 8 | 8 |
| 1 | -2 | -1 | 0 |
| 1 | 0 | 2 | 0 |
| 1 | +2 | -1 | 0 |

The output of the filter (sum) is then scaled according to the following formula:

prog_field[x][y] = sum // 16

The filter aperture can extend outside the active picture area. In this case the pels of the lines outside the active picture should take the value of the closest neighbouring existing pel (below or above) of the same field as defined below.

For all pels [y][x]:

```
if (y<0 && (y&1 == 1)) y=1
if (y<0 && (y&1 == 0)) y=0
if (y>=lower_layer_vertical_size && (y&1 == 1)) y=lower_layer_vertical_size-1
if (y>=lower_layer_vertical_size && (y&1 == 0)) y=lower_layer_vertical_size-2
```

#### 7.7.2.3 Vertical resampling

The frame subject to vertical resampling, input_field, is resampled to the current layer vertical sampling grid using linear interpolation between the sample sites according to the following formula, where output_field is the resulting field:

$$output\_field[x][y_h] = (16 - phase)*input\_field[x][y1] + phase*input\_field[x][y2]$$

| | | | | |
|---|---|---|---|---|
| where | $y_h$ | = | output sample co-ordinate in output field | |
| | y1 | = | $(y_h * m) / n$ | |
| | y2 | = | y1 + 1 | if y1 < lower_layer_vertical_size - 1 |
| | | | y1 | otherwise |
| | phase | = | $16 * ((y_h * m) \% n) // n$ | |
| | m | = | vertical_upsampling_factor_m | |
| | n | = | vertical_upsampling_factor_n | |

#### 7.7.2.4 Horizontal resampling

The frame subject to horizontal resampling, input_field, is resampled to the current layer horizontal sampling grid using linear interpolation between the sample sites according to the following formula, where output_field is the resulting field:

$$output\_field[x_h][y_h] = ((16 - phase)*input\_field[x1][y_h] + phase*input\_field[x2][y_h]) / 256$$

| | | | | |
|---|---|---|---|---|
| where | $x_h$ | = | output sample co-ordinate in output field | |
| | x1 | = | $(x_h * m) / n$ | |
| | x2 | = | x1 + 1 | if y1 < lower_layer_horizontal_size - 1 |
| | | | x1 | otherwise |
| | phase | = | $16 * ((x_h * m) \% n) // n$ | |
| | m | = | horizontal_upsampling_factor_m | |
| | n | = | horizontal_upsampling_factor_n | |

### 7.7.3 Selection and combination of spatial and temporal predictions

The spatial and temporal predictions can be selected or combined to form the actual prediction. The macroblock_type (tables B.2-5, B.2-6 and B.2-7) indicates, by use of the spatial_temporal_weight_class, which can take the values 0, 1, 2, 3, 4, whether the prediction is temporal-only, spatial-only or a weighted combination of temporal and spatial predictions. A fuller description of spatial_temporal_weight_class is given in Section 7.7.4.

In intra pictures, if spatial_temporal_weight_class is 0, normal intra coding is performed, otherwise the prediction is spatial-only. In predicted and interpolated pictures, if the spatial_temporal_weight_class is 0, prediction is temporal-only, if the spatial_temporal_weight_class is 4, prediction is spatial-only, otherwise one or a pair of prediction weights is used to combine the spatial and temporal predictions.

The possible prediction weights are given in a weight table which is selected in the picture scalable extension. Up to four different weight tables are available for use depending on whether the current and lower layers are interlaced or progressive. At the macroblock layer, a one or two bit code, spatial_temporal_weight_code, is used to describe the prediction for each field (or frame), as shown in the table below.

Table showing spatial_temporal_weights and spatial_temporal_weight_classes for the lower_layer_picture_formats and spatial_temporal_weight_codes

| spatial_temporal_weight_ code_table_index | spatial_temporal_weight bits | spatial_temporal_weight code | spatial_temporal_weight (s) | spatial_temporal_weight class | spatial_temporal_integer_weight |
|---|---|---|---|---|---|
| 00 | 1 | 0 | RESERVED | | |
| | 1 | 1 | 0.5 | 1 | 0 |
| 01 | 2 | 00 | 0, 1 | 3 | 1 |
| | 2 | 01 | 0, 0.5 | 1 | 0 |
| | 2 | 10 | 0.5, 1 | 3 | 0 |
| | 2 | 11 | 0.5, 0.5 | 1 | 0 |
| 10 | 2 | 00 | 1, 0 | 2 | 1 |
| | 2 | 01 | 0.5, 0 | 1 | 0 |
| | 2 | 10 | 1, 0.5 | 2 | 0 |
| | 2 | 11 | 0.5, 0.5 | 1 | 0 |
| 11 | 2 | 00 | 1, 0 | 2 | 1 |
| | 2 | 01 | 1, 0.5 | 2 | 0 |
| | 2 | 10 | 0.5, 1 | 3 | 0 |
| | 2 | 11 | 0.5, 0.5 | 1 | 0 |

When the prediction weight combination is given in the form (a,b), "a" gives the proportion of the prediction for the top field which is derived from the spatial prediction and "b" gives the proportion of the prediction for the bottom field which is derived from the spatial prediction for that field.

When the prediction weight combination is given in the form (a), "a" gives the proportion of the prediction for the frame which is derived from the spatial prediction for that frame. The precise method for predictor calculation is as follows:

If pel_pred_temp[y][x] is used to denote the temporal prediction (formed within the curent layer) as defined for pel_pred[y][x] in clause 7.6. pel_pred_lower[y][x] is used to denote the prediction formed from the lower layer, then:

If the weight code is zero then no prediction is made from the lower layer. Therefore;

pel_pred[y][x] = pel_pred_temp[y][x];

If the weight code is one then no prediction is made from the current layer. Therefore;

pel_pred[y][x] = pel_pred_lower[y][x];

If the weight code is one half then the prediction is the average of the temporal and spatial predictions. Therefore;

pel_pred[y][x] = (pel_pred_temp[y][x] + pel_pred_lower[y][x])//2;

When chrominance is sampled 4:2:0, it is treated as interlaced, in accordance with the other prediction modes, that is, the first weight code is used for the top field chrominance lines and the second weight is used for the bottom field chrominance lines.

Note Each field has its own weight code.

It is intended that the different weight tables are used in the following circumstances (although this is not mandatory):

| Lower layer format | Current layer format | spatial_temporal_weight_code_table_index |
|---|---|---|
| Progressive or interlaced | Progressive | 00 |
| Progressive coincident with current layer top fields | Interlaced | 10 |
| Progressive coincident with current layer from bottom fields | Interlaced | 01 |
| Interlaced | Interlaced | 00 or 11 |

### 7.7.4 Updating motion vector predictors and Motion vector selection

In frame pictures where field prediction is used the possibility exists that one of the fields is predicted using spatial-only prediction. In this case no motion vector is present in the bit stream for the field which has spatial-only prediction. For the case where both fields of a frame have spatial-only prediction, the macroblock_type is such that no motion vectors are present in the bit stream for that macroblock.

The class also indicates the number of motion vectors which are present in the coded bit stream and how the prediction vectors are updated and this is described in the Table below. Classes are defined in the following way:

Class 0 indicates temporal-only prediction
Class 1 indicates that neither field has spatial-only prediction
Class 2 indicates that the top field is spatial-only prediction
Class 3 indicates that the bottom field is spatial-only prediction
Class 4 indicates spatial-only prediction.

**Table showing updating of motion vector predictors in Field pictures**

| field motion type | F | B | I | WC | Predictions to update |
|---|---|---|---|---|---|
| Field-based※ | - | - | 1 | 0 | *PMV*[1][0][1:0] = *PMV*[0][0][1:0]_ |
| Field-based | 1 | 1 | 0 | 0 | *PMV*[1][0][1:0] = *PMV*[0][0][1:0]<br>*PMV*[1][1][1:0] = *PMV*[0][1][1:0] |
| Field-based | 1 | 0 | 0 | 0,1 | *PMV*[1][0][1:0] = *PMV*[0][0][1:0] |
| Field-based | 0 | 1 | 0 | 0,1 | *PMV*[1][1][1:0] = *PMV*[0][1][1:0] |
| Field-based※ | 0 | 0 | 0 | 0,1,4 | *PMV*[*r*][*s*][*t*] = 0 § |
| 16x8 MC | 1 | 1 | 0 | 0,1 | (none) |
| 16x8 MC | 1 | 0 | 0 | 0,1 | (none) |
| 16x8 MC | 0 | 1 | 0 | 0,1 | (none) |
| Dual prime | 1 | 0 | 0 | 0 | *PMV*[1][0][1:0] = *PMV*[0][0][1:0] |

Note: *PMV*[*r*][*s*][1:0] = *PMV*[*u*][*v*][1:0] means that;
*PMV*[*r*][*s*][1] = *PMV*[*u*][*v*][1] and *PMV*[*r*][*s*][0] = *PMV*[*u*][*v*][0]

_ If **concealment_motion_vectors** is zero then *PMV*[*r*][*s*][*t*] is set to zero (for all *r*, *s* and *t*).

※ **field_motion_type** is not present in the bitstream but is assumed to be Field-based

§ In P-pictures, *PMV*[*r*][*s*][*t*] is set to zero (for all *r*, *s* and *t*). See clause 7.6.3.4. The *PMV*[*r*][*s*][*t*] is not reset in B-pictures

*Comment*: WC means spatial_temporal_weight_class. This table is valid for non-scalable coding and for spatial scalability.

**Table showing updating of motion vector predictors in Frame pictures**

| frame motion type | F | B | I | WC | Predictions to update |
|---|---|---|---|---|---|
| Frame-based※ | - | - | 1 | 0 | PMV[1][0][1:0] = PMV[0][0][1:0] |
| Frame-based | 1 | 1 | 0 | 0 | PMV[1][0][1:0] = PMV[0][0][1:0]<br>PMV[1][1][1:0] = PMV[0][1][1:0] |
| Frame-based | 1 | 0 | 0 | 0,1,2,3 | PMV[1][0][1:0] = PMV[0][0][1:0] |
| Frame-based | 0 | 1 | 0 | 0,1,2,3 | PMV[1][1][1:0] = PMV[0][1][1:0] |
| Frame-based※ | 0 | 0 | 0 | 0,1,2,3, 4 | PMV[r][s][t] = 0§ |
| Field-based | 1 | 1 | 0 | 0 | (none) |
| Field-based | 1 | 0 | 0 | 0,1 | (none) |
| Field-based | 1 | 0 | 0 | 2 | PMV[0][0][1:0] = PMV[1][0][1:0] |
| Field-based | 1 | 0 | 0 | 3 | PMV[1][0][1:0] = PMV[0][0][1:0] |
| Field-based | 0 | 1 | 0 | 0,1 | (none) |
| Field-based | 0 | 1 | 0 | 2 | PMV[0][1][1:0] = PMV[1][1][1:0] |
| Field-based | 0 | 1 | 0 | 3 | PMV[1][1][1:0] = PMV[0][1][1:0] |
| Dual prime@ | 1 | 0 | 0 | 0,2,3 | PMV[1][0][1:0] = PMV[0][0][1:0] |

Note: *PMV*[*r*][*s*][1:0] = *PMV*[*u*][*v*][1:0] means that;

*PMV*[*r*][*s*][1] = *PMV*[*u*][*v*][1] and *PMV*[*r*][*s*][0] = *PMV*[*u*][*v*][0]

_ If **concealment_motion_vectors** is zero then *PMV*[*r*][*s*][*t*] is set to zero (for all *r*, *s* and *t*).

※ **frame_motion_type** is not present in the bitstream but is assumed to be Frame-based

§ In P-pictures *PMV*[*r*][*s*][*t*] is set to zero (for all *r*, *s* and *t*). See clause 7.6.3.4. The *PMV*[*r*][*s*][*t*] is not reset in B-pictures

@ Dual prime can not be used when spatial_temporal_integer_weight = "0".

*Comment*: WC means spatial_temporal_weight_class. This table is valid for non-scalable coding and for spatial scalability.

Table showing predictions and vectors in Field pictures

| field motion type | F | B | I | WC | Vector | Prediction formed for |
|---|---|---|---|---|---|---|
| Field-based※ | - | - | 1 | 0 | vector[0][0][1:0]_ | None (vector is for concealment) |
| Field-based | 1 | 1 | 0 | 0 | vector[0][0][1:0]<br>vector[0][1][1:0] | whole field, forward<br>whole field, backward |
| Field-based | 1 | 0 | 0 | 0,1 | vector[0][0][1:0] | whole field, forward |
| Field-based | 0 | 1 | 0 | 0,1 | vector[0][1][1:0] | whole field, backward |
| Field-based※ | 0 | 0 | 0 | 0,1,4 | vector[0][0][1:0]*§ | whole field, forward |
| 16x8 MC | 1 | 1 | 0 | 0,1 | vector[0][0][1:0]<br>vector[1][0][1:0]<br>vector[0][1][1:0]<br>vector[1][1][1:0] | upper 16x8 field, forward<br>lower 16x8 field, forward<br>upper 16x8 field, backward<br>lower 16x8 field, backward |
| 16x8 MC | 1 | 0 | 0 | 0,1 | vector[0][0][1:0]<br>vector[1][0][1:0] | upper 16x8 field, forward<br>lower 16x8 field, forward |
| 16x8 MC | 0 | 1 | 0 | 0,1 | vector[0][1][1:0]<br>vector[1][1][1:0] | upper 16x8 field, backward<br>lower 16x8 field, backward |
| Dual prime | 1 | 0 | 0 | 0 | vector[0][0][1:0]<br>vector[2][0][1:0]*※ | same parity whole field, forward<br>opposite parity whole field, forward |

Note: Vectors are listed in the order they appear in the bitstream

_ the vector is only present if **concealment_motion_vectors** is one

※ **field_motion_type** is not present in the bitstream but is assumed to be Field-based

* These vectors are not present in the bitstream

※ These vectors are derived from *vector*[0][0][1:0] as described in clause 7-?

§ The vector is taken to be (zero, zero) as explained in clause 7-?

*Comment*: WC means spatial_temporal_weight_class. This table is valid for non-scalable coding and for spatial scalability.

**Table showing predictions and vectors in Frame pictures**

| frame motion type | F | B | I | WC | Vector | Prediction formed for |
|---|---|---|---|---|---|---|
| Frame-based▒ | - | - | 1 | 0 | *vector*[0][0][1:0]_ | None (vector is for concealment) |
| Frame-based | 1 | 1 | 0 | 0 | *vector*[0][0][1:0]<br>*vector*[0][1][1:0] | frame, forward<br>frame, backward |
| Frame-based | 1 | 0 | 0 | 0,1,2,3 | *vector*[0][0][1:0] | frame, forward |
| Frame-based | 0 | 1 | 0 | 0,1,2,3 | *vector*[0][1][1:0] | frame, backward |
| Frame-based▒ | 0 | 0 | 0 | 0,1,2,3, 4 | *vector*[0][0][1:0]*§ | frame, forward |
| Field-based | 1 | 1 | 0 | 0 | *vector*[0][0][1:0]<br>*vector*[1][0][1:0]<br>*vector*[0][1][1:0]<br>*vector*[1][1][1:0] | top field, forward<br>bottom field, forward<br>top field, backward<br>bottom field, backward |
| Field-based | 1 | 0 | 0 | 0,1 | *vector*[0][0][1:0]<br>*vector*[1][0][1:0] | top field, forward<br>bottom field, forward |
| Field-based | 1 | 0 | 0 | 2 | <br>*vector*[1][0][1:0] | top field, spatial<br>bottom field, forward |
| Field-based | 1 | 0 | 0 | 3 | *vector*[0][0][1:0] | top field, forward<br>bottom field, spatial |
| Field-based | 0 | 1 | 0 | 0,1 | *vector*[0][1][1:0]<br>*vector*[1][1][1:0] | top field, backward<br>bottom field, backward |
| Field-based | 0 | 1 | 0 | 2 | <br>*vector*[1][1][1:0] | top field, spatial<br>bottom field, backward |
| Field-based | 0 | 1 | 0 | 3 | *vector*[0][1][1:0] | top field, backward<br>bottom field, spatial |
| Dual prime@ | 1 | 0 | 0 | 0,2,3 | *vector*[0][0][1:0]<br>*vector*[0][0][1:0]*<br>*vector*[2][0][1:0]*▒<br>*vector*[3][0][1:0]*▒ | same parity top field, forward<br>same parity bottom field, forward<br>opposite parity top field, forward<br>opposite parity bottom field, forward |

Note: Vectors are listed in the order they appear in the bitstream

_ the vector is only present if **concealment_motion_vectors** is one

▒ **frame_motion_type** is not present in the bitstream but is assumed to be Frame-based

* These vectors are not present in the bitstream

▒ These vectors are derived from *vector*[0][0][1:0] as described in clause 7-?

§ The vector is taken to be (zero, zero) as explained in clause 7-?

@ Dual prime can not be used when spatial_temporal_integer_weight = "0".

*Comment*: WC means spatial_temporal_weight_class. This table is valid for non-scalable coding and for spatial scalability.

### 7.7.5 Skipped macroblocks

In all cases, a skipped macroblock is only the result of a prediction, and all the DCT coefficients are considered to be zero.

The following rules apply for non-scalable bit streams; i.e. if sequence_scalable_extension is not present.

In I-frame pictures or field pictures, if sequence_scalable_extension is not present, then all macroblocks are coded and there are no skipped macroblocks; i.e. The syntax element *macroblock_address_increment* is always equal to "1", accept for the first macroblock of a slice, in which case it indicates the horizontal position of the slice.

In P-frame pictures a skipped macroblock is defined to be a frame-based predicted macroblock with a reconstructed frame motion vector equal to zero.

In P-field pictures, a skipped macroblock is defined to be a field-based predicted macroblock with a reconstructed field motion vector equal to zero. The reference field for the prediction is the same parity field.

In B frame pictures, a skipped macoblock is defined to be a frame-based predicted macroblock with differential frame motion vector(s) equal to zero. The type of prediction (forward, backward or averaged) is the same as the prior macroblock.

In B field pictures, a skipped macoblock is defined to be a field-based predicted macroblock with differential field motion vector(s) equal to zero. The type of prediction (forward, backward or averaged) is the same as the prior macroblock. A reference field(s) for the prediction is (are) the same parity field(s).

In B-frame or B-field pictures, a skipped macroblock shall not follow an intra-coded macroblock.

If sequence_scalable_extension is present and scalable_mode = spatial_scalablity, the following rules apply in addition to those given above.

In I-frame or I-field pictures, skipped macroblocks are allowed. These are defined as spatial-only predicted.
In P-pictures and B-pictures, the skipped macroblock is temporal-only predicted.

Table for B.2-5

Macroblock type table for intra pictures in spatial scalability

| MB_type VLC code | Q | F MV | B MV | C B P | I n t r a | spatial_temporal_weight_code_flag | Description | spatial_temporal_weight_classes allowed |
|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | Coded, Compatible | 4 |
| 01 | 1 | 0 | 0 | 1 | 0 | 0 | Coded, Compatible, Quant | 4 |
| 0011 | 0 | 0 | 0 | 0 | 1 | 0 | Intra | 0 |
| 0010 | 1 | 0 | 0 | 0 | 1 | 0 | Intra, Quant | 0 |
| 0001 | 0 | 0 | 0 | 0 | 0 | 0 | Not Coded, Compatible | 4 |

*Information*: spatial_temporal_weight_codes are not present in the bitstream in intra pictures. Skipped macroblocks are spatial-only predicted.

Table for B.2-6

Macroblock type table for predicted pictures in spatial scalability

| MB_type VLC code | Q | F MV | B MV | C B P | I n t r a | spatial_temporal_weight_code_flag | Description | spatial_temporal_weight_classes allowed |
|---|---|---|---|---|---|---|---|---|
| 10 | Q | 1 | 0 | 1 | 0 | 0 | MC, Coded | 0 |
| 011 | 0 | 1 | 0 | 1 | 0 | 1 | MC, Coded, Compatible | 1,2,3 |
| 0000 100 | 0 | 0 | 0 | 1 | 0 | 0 | No MC, Coded | 0 |
| 0001 11 | 0 | 0 | 0 | 1 | 0 | 1 | No MC, Coded, Compatible | 1,2,3 |
| 0010 | 0 | 1 | 0 | 0 | 0 | 0 | MC, Not Coded | 0 |
| 0000 111 | 0 | 0 | 0 | 0 | 1 | 0 | Intra | 0 |
| 0011 | 0 | 1 | 0 | 0 | 0 | 1 | MC, Not coded, Compatible | 1,2,3 |
| 010 | 1 | 1 | 0 | 1 | 0 | 0 | MC, Coded, Quant | 0 |
| 0001 00 | 1 | 0 | 0 | 1 | 0 | 0 | No MC, Coded, Quant | 0 |
| 0000 110 | 1 | 0 | 0 | 0 | 1 | 0 | Intra, Quant | 0 |
| 11 | 1 | 1 | 0 | 1 | 0 | 1 | MC, Coded, Compatible, Quant | 1,2,3 |
| 0001 01 | 1 | 0 | 0 | 1 | 0 | 1 | No MC, Coded, Compatible, Quant | 1,2,3 |
| 0001 10 | 0 | 0 | 0 | 0 | 0 | 1 | No MC, Not Coded, Compatible | 1,2,3 |
| 0000 101 | 0 | 0 | 0 | 1 | 0 | 0 | Coded, Compatible | 4 |
| 0000 010 | 1 | 0 | 0 | 1 | 0 | 0 | Coded, Compatible, Quant | 4 |
| 0000 011 | 0 | 0 | 0 | 0 | 0 | 0 | Not Coded, Compatible | 4 |

*Information*: spatial_temporal_weight_codes are only present in the bitstream when the spatial_temporal_weight_class is non zero. Skipped macroblocks are temporal-only predicted.

Table for B.2-7

Macroblock type table for interpolated pictures in spatial scalability

| MB_type VLC code | Q | F M V | B M V | C B P | I n t r a | spatial_temporal _weight_code _flag | Description | spatial_ temporal _weight_ classes allowed |
|---|---|---|---|---|---|---|---|---|
| 10 | 0 | 1 | 1 | 0 | 0 | 0 | Interp, Not coded | 0 |
| 11 | 0 | 1 | 1 | 1 | 0 | 0 | Interp, Coded | 0 |
| 010 | 0 | 0 | 1 | 0 | 0 | 0 | Back, Not coded | 0 |
| 011 | 0 | 0 | 1 | 1 | 0 | 0 | Back, Coded | 0 |
| 0010 | 0 | 1 | 0 | 0 | 0 | 0 | For, Not coded | 0 |
| 0011 | 0 | 1 | 0 | 1 | 0 | 0 | For, Coded | 0 |
| 0001 10 | 0 | 0 | 1 | 0 | 0 | 1 | Back, Not Coded, Compatible | 1,2,3 |
| 0001 11 | 0 | 0 | 1 | 1 | 0 | 1 | Back, Coded, Compatible | 1,2,3 |
| 0001 00 | 0 | 1 | 0 | 0 | 0 | 1 | For, Not Coded, Compatible | 1,2,3 |
| 0001 01 | 0 | 1 | 0 | 1 | 0 | 1 | For, Coded, Compatible | 1,2,3 |
| 0000 110 | 0 | 0 | 0 | 0 | 1 | 0 | Intra | 0 |
| 0000 111 | 1 | 1 | 1 | 1 | 0 | 0 | Interp, Coded, Quant | 0 |
| 0000 100 | 1 | 1 | 0 | 1 | 0 | 0 | For, Coded, Quant | 0 |
| 0000 101 | 1 | 0 | 1 | 1 | 0 | 0 | Back, Coded, Quant | 0 |
| 0000 0100 | 1 | 0 | 0 | 0 | 1 | 0 | Intra, Quant | 0 |
| 0000 0101 | 1 | 1 | 0 | 1 | 0 | 1 | For, Coded, Compatible, Quant | 1,2,3 |
| 0000 0110 0 | 1 | 0 | 1 | 1 | 0 | 1 | Back, Coded, Compatible, Quant | 1,2,3 |
| 0000 0111 0 | 0 | 0 | 0 | 0 | 0 | 0 | Not Coded, Compatible | 4 |
| 0000 0110 1 | 1 | 0 | 0 | 1 | 0 | 0 | Coded, Compatible, Quant | 4 |
| 0000 0111 1 | 0 | 0 | 0 | 1 | 0 | 0 | Coded, Compatible | 4 |

*Information*: spatial_temporal_weight_codes are only present in the bitstream when the spatial_temporal_weight_class is non zero. Skipped macroblocks are temporal-only predicted.

End of document