



JCTVC-P0135/JCT3V-G0135:
MV-HEVC/SHVC HLS:
Auxiliary pictures for multiple overlays

Jill Boyce and Stephan Wenger

Introduction

- Related to contribution JCTVC-O0358 (and JCTVC-P0092) to support overlays
- Use auxiliary pictures
- Extend for greater flexibility, including support for multiple overlays per primary picture

Proposal overview

1. Signal an `aux_type` syntax element in the VPS for each value of `AuxId` greater than 0, rather than inferring type from the `AuxId` value
2. Define 3 auxiliary picture types:
 - Overlay content
 - Overlay layout
 - Overlay alpha
3. New overlay info SEI message

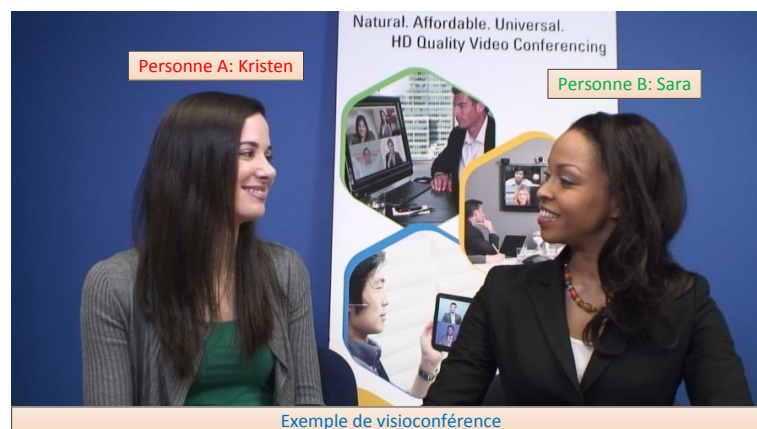
Example



Figure 1. Primary picture



2(a)



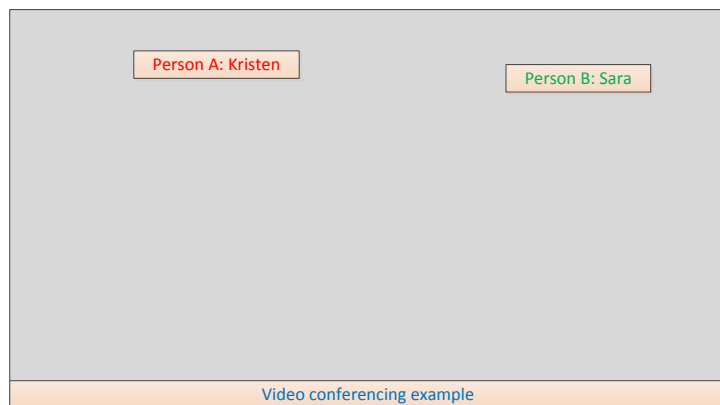
2(b)

Figure 2. Resulting display with (a) English overlay (b) French overlay, with all elements displayed

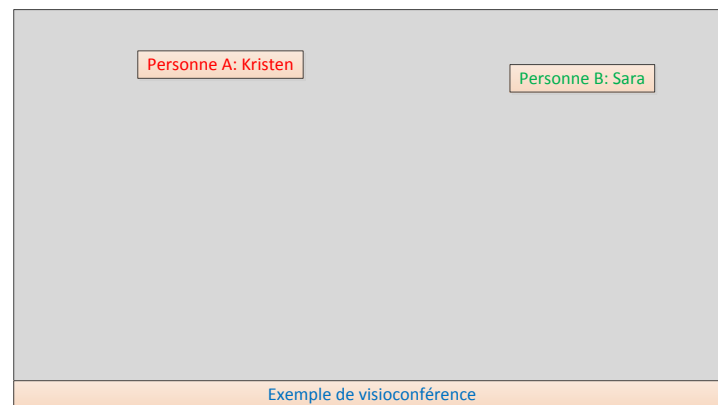
Example continued



Figure 3. Overlay layout



4(a)



4(b)

Figure 4. Overlay content for (a) English overlay (b) French overlay

1. Signal aux type separately from AuxId

- In current design, aux type determined by AuxId
 - Restricts a single auxiliary picture of a particular type for a given primary picture
- Propose to separately signal aux_type in VPS extension
 - Would affect all aux pic types, not just overlay types
 - Allows more than one auxiliary picture of a particular type to be associated with same primary picture
- Each auxiliary picture layer has own nuh_layer_id value
 - Associated primary picture layer determined by the ViewId and DependencyId values, as determined by the ScalabilityId
 - AuxId distinguishes between aux pic layers
 - aux_type defines auxiliary picture type

1. Signal aux type separately from AuxId

| | |
|-------------------------------------|------|
| vps_extension() | |
| ... | |
| for(i = 1; i < NumAuxLayers; i++) | |
| aux_type[i] | u(8) |
| } | |
| ... | |

NumViews = 1

NumAuxLayers = 0

```

for( i = 0; i <= MaxLayersMinus1; i++ ) {
    lId = layer_id_in_nuh[ i ]
    for( smIdx= 0, j = 0; smIdx < 16; smIdx++ )
        if( scalability_mask_flag[ smIdx ] )
            ScalabilityId[ i ][ smIdx ] = dimension_id[ i ][ j++ ]
    ViewOrderIdx[ lId ] = ScalabilityId[ i ][ 1 ]
    if( i > 0 && ( ViewOrderIdx[ lId ] != ScalabilityId[ i - 1 ][ 1 ] ) )
        NumViews++
    ViewScalExtLayerFlag[ lId ] = ( ViewOrderIdx[ lId ] > 0 )
    AuxId[ lId ] = ScalabilityId[ i ][ 3 ]
    if( i > 0 && ( AuxId[ lId ] != ScalabilityId[ i - 1 ][ 3 ] ) )
        NumAuxLayers++
}

```

2. Define 3 new aux types

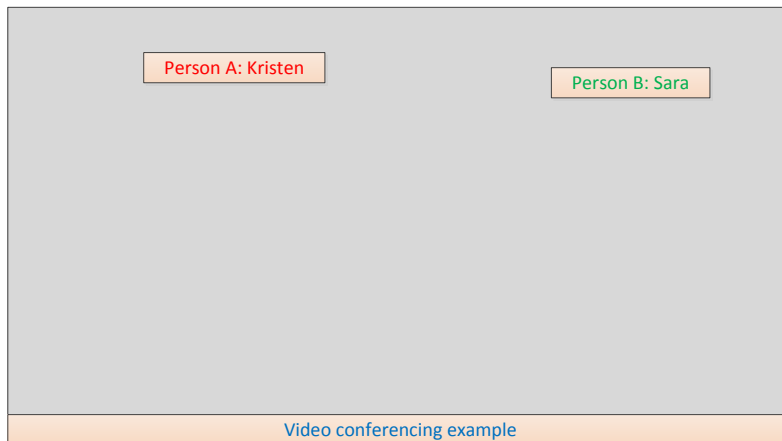
- Define 3 auxiliary picture types:
 - Overlay content
 - Overlay layout
 - Overlay alpha

Table F-1 – Mapping of `aux_type[i]` to the type of auxiliary pictures

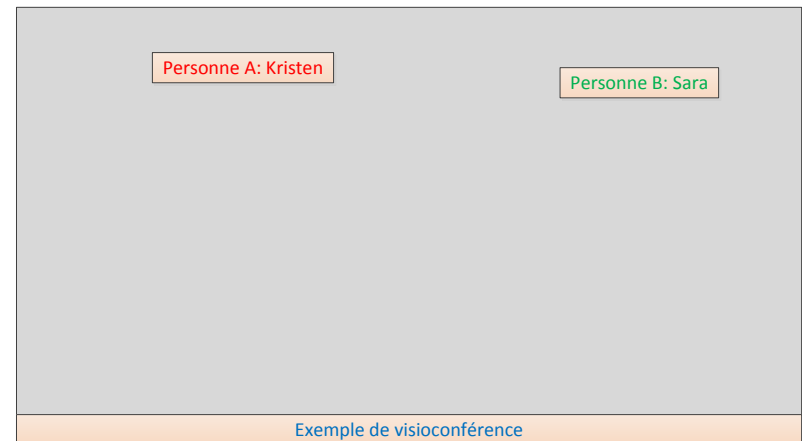
| <code>aux_type[i]</code> | Name of <code>aux_type[i]</code> | Type of auxiliary pictures |
|----------------------------|------------------------------------|--|
| 0 | AUX_ALPHA | Alpha plane of non-auxiliary picture |
| 1 | AUX_DEPTH | Depth picture |
| 2 | AUX_OVERLAY_CONTENT | Overlay content picture |
| 3 | AUX_LAYOUT | Layout of auxiliary overlay picture |
| 4 | AUX_OVERLAY_ALPHA | Alpha plane of auxiliary overlay picture |
| 5 -127 | | Reserved |
| 128-143 | | Unspecified |
| 144-255 | | Reserved |

Aux pic type: Overlay content

- Contains overly content, intended for display as an overlay over the associated primary picture
- Multiple overlay content aux pics may be associated with the same primary picture
 - For example, for different languages



4(a)



4(b)

Figure 4. Overlay content for (a) English overlay (b) French overlay

Aux pic type: Overlay layout

- Describes a mapping of the picture into multiple overlay elements
- May be shared by multiple overlays
- Parameters in the overlay info SEI message are used to interpret content
 - Coded sample values determine element selection
 - Min and max sample values for each element signaled in SEI
- Overlay layout pictures may use a lower frame rate than overlay content



Aux pic type: Overlay alpha

- Apply alpha to the overlay content picture, rather than to the primary picture
- May be shared by multiple overlays
- Optional

3. Overlay info SEI message

- Defines associations between the auxiliary picture layers used for defining an overlay
- Describes contents of the overlay layout
 - How regions of the picture are divided into individually controllable overlay elements
- Persistence of entire message
- Persistence of overlay layout
- Persistence of overlay alpha

Associations of aux pic layers with overlays

- Number of overlays
- For each overlay
 - Layer id of each of
 - Overlay content
 - Overlay layout (optional)
 - Overlay alpha (optional)

Example 1: Two overlays with shared overlay layout

- Bitstream has a single view, with two overlays, and a shared overlay layout

Signaled in VPS extension:

| Description | LayerId | View OrderIdx | AuxId | AuxType |
|-----------------------------|---------|---------------|-------|---------|
| Base view primary pic | 0 | 0 | 0 | 0 |
| Base view overlay layout | 1 | 0 | 1 | 4 |
| Base view overlay content A | 2 | 0 | 2 | 3 |
| Base view overlay content B | 3 | 0 | 3 | 3 |

Signaled in SEI:

num_overlays_minus1 = 1 (2 overlays)

| overlay_idx | overlay_content_layer_id | overlay_layout_layer_id | overlay_alpha_layer_id |
|-------------|--------------------------|-------------------------|------------------------|
| 0 | 2 | 1 | - |
| 1 | 3 | 1 | - |

Example 2: Two views with two overlays

- Bitstream contains two views, each of which has two overlays corresponding to two languages. Each overlay has a unique overlay layout and overlay alpha.

| Description | LayerId | View OrderIdx | AuxId | AuxType |
|---|---------|---------------|-------|---------|
| Base view primary pic | 0 | 0 | 0 | - |
| Enhanced view primary pic | 1 | 1 | 0 | - |
| Base view overlay A content (English) | 2 | 0 | 1 | 3 |
| Base view overlay A layout | 3 | 0 | 2 | 4 |
| Base view overlay A alpha | 4 | 0 | 3 | 5 |
| Base view overlay B content (French) | 5 | 0 | 4 | 3 |
| Base view overlay B layout | 6 | 0 | 5 | 4 |
| Base view overlay B alpha | 7 | 0 | 6 | 5 |
| Enhanced view overlay A content (English) | 8 | 1 | 1 | 3 |
| Enhanced view overlay A layout | 9 | 1 | 2 | 4 |
| Enhanced view overlay A alpha | 10 | 1 | 3 | 5 |
| Enhanced view overlay B content (French) | 11 | 1 | 4 | 3 |
| Enhanced view overlay B layout | 12 | 1 | 5 | 4 |
| Enhanced view overlay B alpha | 13 | 1 | 6 | 5 |

| overlay_idx | overlay_content_layer_id | overlay_layout_layer_id | overlay_alpha_layer_id |
|-------------|--------------------------|-------------------------|------------------------|
| 0 | 2 | 3 | 4 |
| 1 | 5 | 6 | 7 |
| 2 | 8 | 9 | 10 |
| 3 | 11 | 12 | 13 |

Example 3: Two views with overlay and alpha

- Bitstream contains two views, each of which has two overlays. Each view's overlay has a single overlay layout and alpha, shared by the two language overlays.

| Description | LayerId | View OrderIdx | AuxId | AuxType |
|--|---------|---------------|-------|---------|
| Base view primary pic | 0 | 0 | 0 | - |
| Enhan view primary pic | 1 | 1 | 0 | - |
| Base view overlay A content (English) | 2 | 0 | 1 | 3 |
| Base view overlay layout | 3 | 0 | 2 | 4 |
| Base view overlay alpha | 4 | 0 | 3 | 5 |
| Base view overlay B content (French) | 5 | 0 | 1 | 3 |
| Enhan view overlay A content (English) | 6 | 1 | 1 | 3 |
| Enhan view overlay layout | 7 | 1 | 2 | 4 |
| Enhan view overlay alpha | 8 | 1 | 3 | 5 |
| Enhan view overlay B content (French) | 9 | 1 | 1 | 3 |

| overlay_idx | overlay_content_layer_id | overlay_layout_layer_id | overlay_alpha_layer_id |
|-------------|--------------------------|-------------------------|------------------------|
| 0 | 2 | 3 | 4 |
| 1 | 5 | 3 | 4 |
| 2 | 6 | 7 | 8 |
| 3 | 9 | 7 | 8 |

Example 4: Two scalable coded overlays

- Bitstream contains two scalable coded layers. Two overlays are scalable coded, with each of the overlay content, overlay layout and overlay alpha scalable coded.

| Description | LayerId | DependencyId | AuxId | AuxType | direct dependent reference layer |
|------------------------------|---------|--------------|-------|---------|----------------------------------|
| Base layer primary pic | 0 | 0 | 0 | - | - |
| Enhanc layer primary pic | 1 | 1 | 0 | - | 0 |
| Base layer overlay content | 2 | 0 | 1 | 3 | - |
| Base layer overlay layout | 3 | 0 | 2 | 4 | - |
| Base layer overlay alpha | 4 | 0 | 3 | 5 | - |
| Enhanc layer overlay content | 5 | 1 | 1 | 3 | 2 |
| Enhanc layer overlay layout | 6 | 1 | 2 | 4 | 3 |
| Enhanc layer overlay alpha | 7 | 1 | 3 | 5 | 4 |

| overlay_idx | overlay_content_layer_id | overlay_layout_layer_id | overlay_alpha_layer_id |
|-------------|--------------------------|-------------------------|------------------------|
| 0 | 2 | 1 | 4 |
| 1 | 5 | 6 | 7 |

Mapping of overlay layout into elements

- In overlay info SEI message
- For each overlay layout, signal number of elements
- For each element, signal min and max sample values
- For each sample position, element index mapping is dependent on sample value, and which min and max values it falls between

Overlay SEI message syntax

| | |
|--|-------|
| overlay_info() { | |
| overlay_info_cancel_flag | u(1) |
| if (!overlay_info_cancel_flag) { | |
| num_overlays_minus1 | ue(v) |
| overlay_info_name_len | ue(v) |
| for(i = 0; i < num_overlays_minus1; i++) { | |
| overlay_idx[i] | ue(v) |
| overlay_name[i] | f(v) |
| overlay_content_layer_id[i] | u(6) |
| overlay_layout_present_flag[i] | u(1) |
| if (overlay_layout_present_flag[i]) { | |
| overlay_layout_layer_id[i] | u(6) |
| overlay_layout_persistence_flag[i] | u(1) |
| } | |
| overlay_alpha_present_flag[i] | u(1) |
| if (overlay_alpha_present_flag[i]) { | |
| overlay_alpha_layer_id[i] | u(6) |
| overlay_alpha_persistence_flag[i] | u(1) |
| } | |
| } | |
| if (overlay_layout_present_flag[i]) { | |
| num_overlay_elements_minus1[i] | ue(v) |
| for(j = 0; j <= num_overlay_elements_minus1[i]; j++) { | |
| overlay_element_idx[i][j] | ue(v) |
| overlay_element_name[i][j] | f(v) |
| overlay_element_layout_min[i][j] | u(v) |
| overlay_element_layout_max[i][j] | u(v) |
| } | |
| } | |
| overlay_info_persistence_flag | u(1) |
| } | |
| } | |

Additional features supported vs. JCTVC-00358

- Multiple overlays per primary picture layer
- Explicit indication of association of auxiliary pictures with primary picture layers (views, scalable layers)
- Shared or scalable coding of overlay auxiliary picture content across primary picture layers
- Shared or non-shared use of overlay layout map and overlay alpha for multiple overlays
- Individually controllable persistence indicators for overlay layout map and overlay alpha, to enable use case where frame rate is lower than overlay content