



Proposal for Supporting Optional Overlays with Help of Auxiliary Pictures

**Niko Stefanoski, Oliver Wang,
Aljosa Smolic, Ted Szypulski**

Disney Research Zurich
ESPN





- Proposed to use auxiliary pictures to enable Optional Overlays functionality
- This proposal is based on JCTVC-O0358/ JCT3V-F0057 that proposed to use views to represent overlay data
- Auxiliary pictures mechanism adjusted
 - aux picture types can be specified in SEI message





What are Optional Overlays?





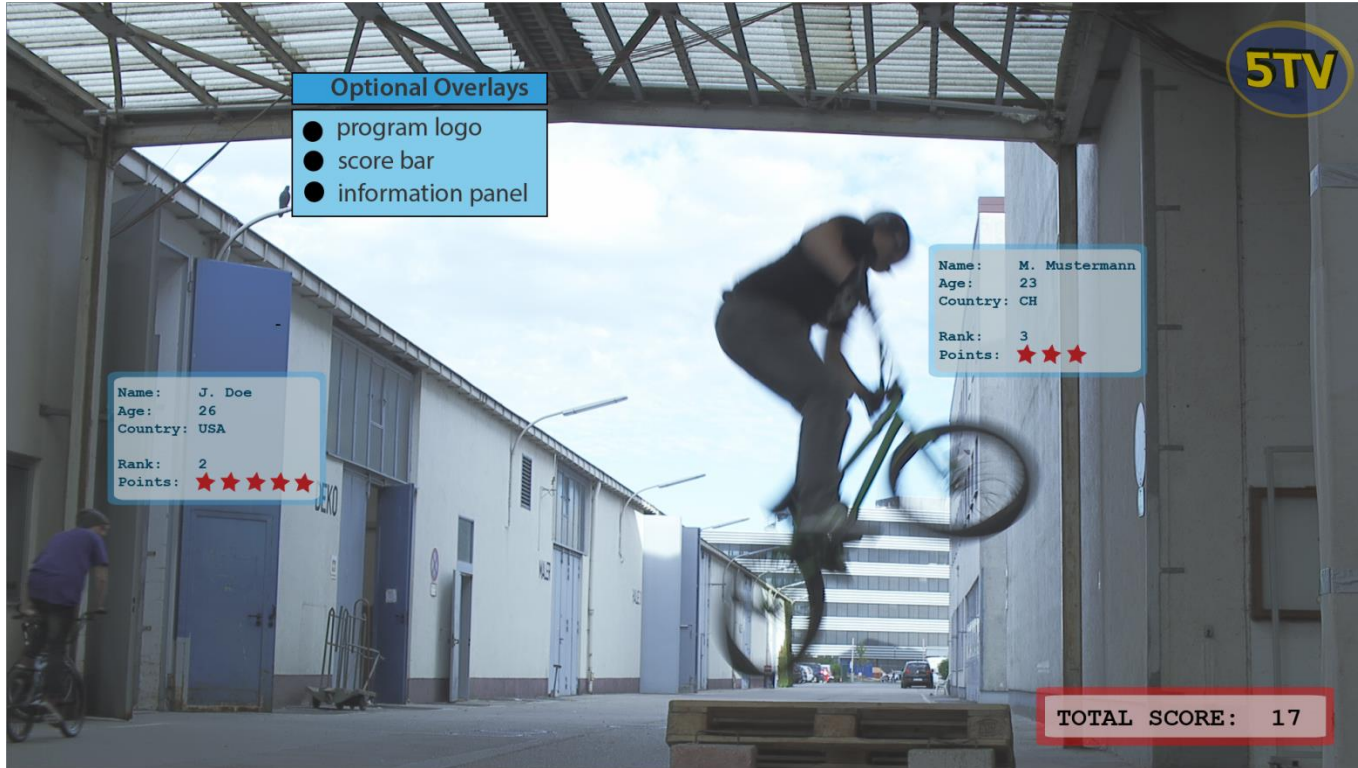


What are Optional Overlays?





What are Optional Overlays?



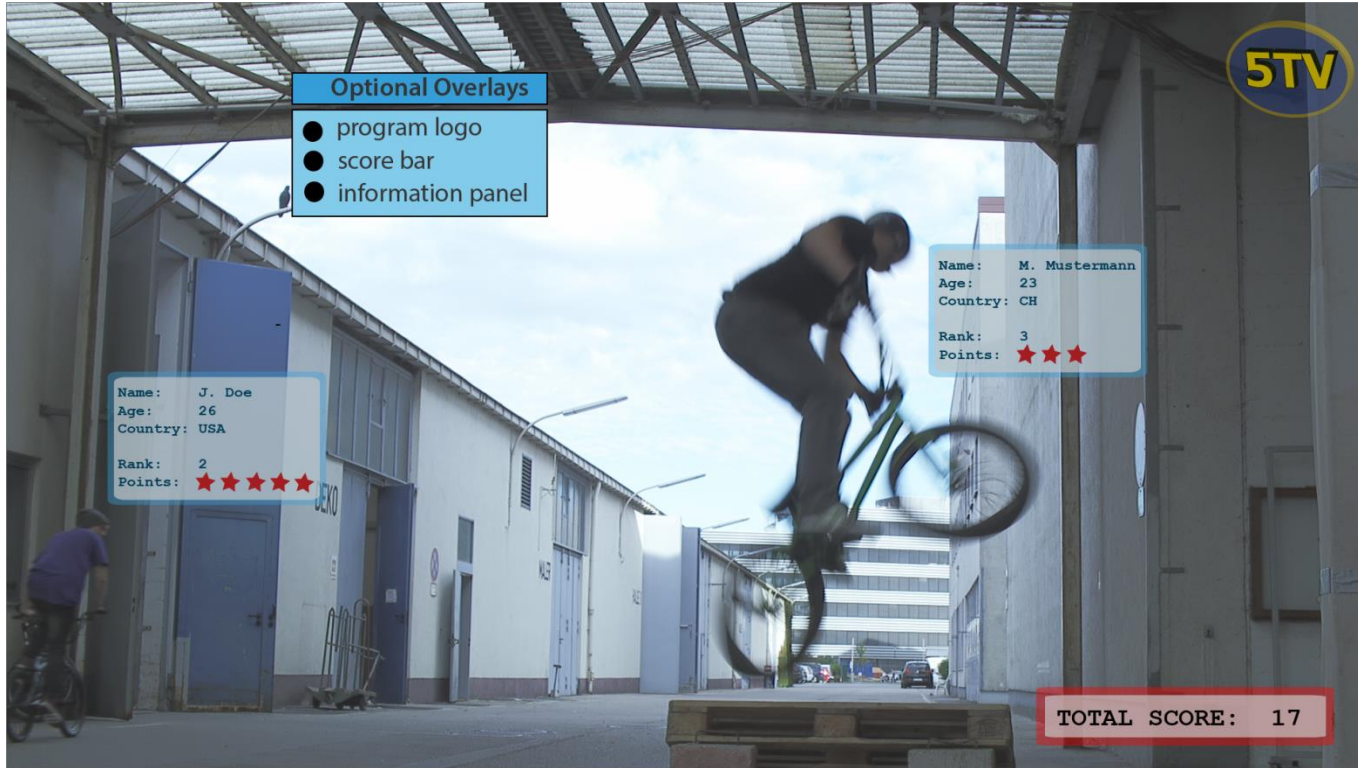


What are Optional Overlays?





What are Optional Overlays?





What are Optional Overlays?





What are Optional Overlays?





Proposal

- Realize Optional Overlays functionality the auxiliary pictures mechanism
- Represent overlay data in separate auxiliary picture layers
- Define SEI message to
 - signal availability of Optional Overlays functionality
 - provide additional parameters required to realize the functionality





Video and Overlay Data

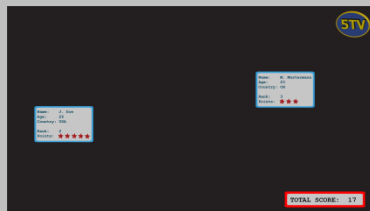
2D Video Data

video pictures, YUV

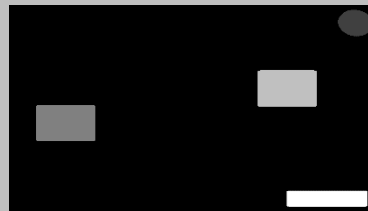


Overlay Data

overlay pictures, YUV



label pictures, Y



alpha pictures, Y



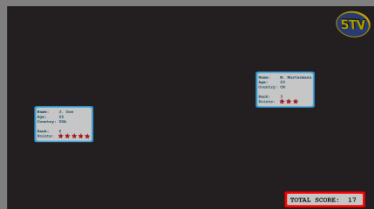


Video and Overlay Data

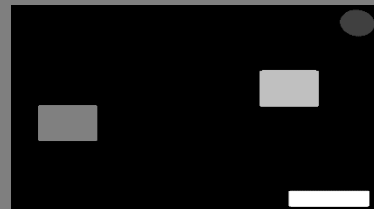
Primary layer
video pictures, YUV



Aux layer 0 to K-1
overlay pictures, YUV



Aux layer K
label pictures, Y



Aux layer K+1
alpha pictures, Y





SEI Message Syntax and Semantics

	Descriptor
optional_overlays_info (payloadSize) {	
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Specifies
persistence of SEI
message



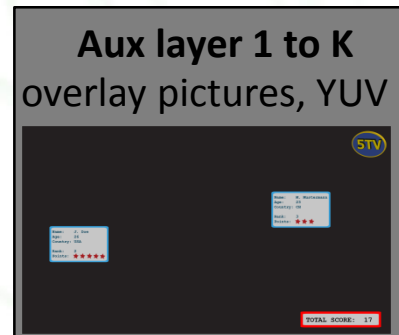


SEI Message Syntax and Semantics

optional_overlays_info (payloadSize) {	Descriptor
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Number of
different layers
containing overlay
pictures

Example:





SEI Message Syntax and Semantics

optional_overlays_info (payloadSize) {	Descriptor
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Specifies the Auxlds of all layers containing overlay pictures

Example:



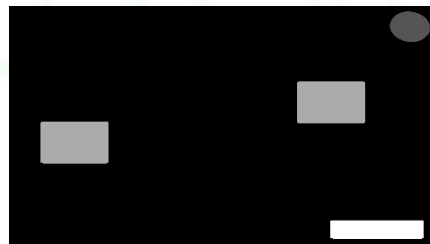


SEI Message Syntax and Semantics

optional_overlays_info (payloadSize) {	Descriptor
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Specifies the AuxId of the layer containing label pictures

Example:





SEI Message Syntax and Semantics

optional_overlays_info (payloadSize) {	Descriptor
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Specifies if alpha pictures are present



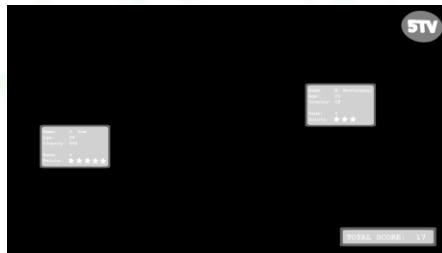


SEI Message Syntax and Semantics

optional_overlays_info (payloadSize) {	Descriptor
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Specifies the AuxId of the layer containing alpha pictures

Example:



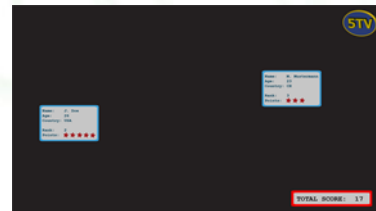


SEI Message Syntax and Semantics

optional_overlays_info (payloadSize) {	Descriptor
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Number of overlay elements within a overlay picture

Example:





SEI Message Syntax and Semantics

	Descriptor
optional_overlays_info (payloadSize) {	
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Names for all
overlay elements in
all overlay pictures

Example

Optional Overlays

- ☐ program logo
- ☐ score bar
- ☐ information panel



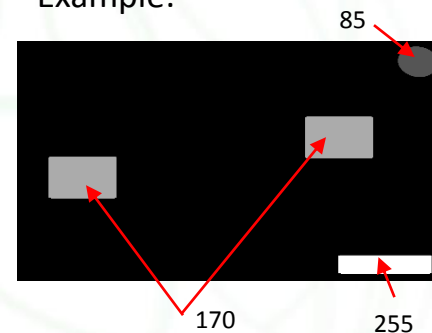


SEI Message Syntax and Semantics

<code>optional_overlays_info (payloadSize) {</code>	Descriptor
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

Numerical identifiers for overlay elements within a label picture

Example:





SEI Message Syntax and Semantics

optional_overlays_info (payloadSize) {	Descriptor
oov_info_cancel_flag	u(1)
if(oov_info_cancel_flag == 0) {	
oov_ol_layer_cnt_minus1	u(4)
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_auxid_ol_minus128[j]	u(4)
}	
oov_auxid_label_minus128	u(4)
oov_alpha_present_flag	u(1)
if(oov_alpha_present_flag == 1) {	
oov_auxid_alpha_minus128	u(4)
}	
oov_ol_elem_cnt_minus1	ue(v)
for (i=0; i <= oov_ol_elem_cnt_minus1; i++) {	
for (j=0; j <= oov_ol_layer_cnt_minus1; j++)	
oov_ol_elem_name[i][j]	f(512)
oov_label_id[i]	u(v)
}	
oov_label_offset	u(v)
}	

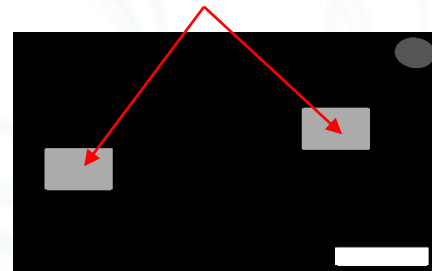
Used to specify a range of label values which identify an overlay element

Example:

oov_label_offset = 32

oov_label_id = 170

$$170-32 \leq x \leq 170+32$$





User Selection

Decoded data:

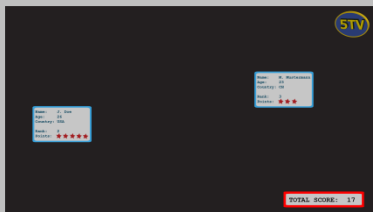
2D Video Data

resSamples_{Vid}

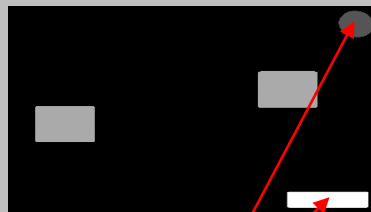


Overlay Data

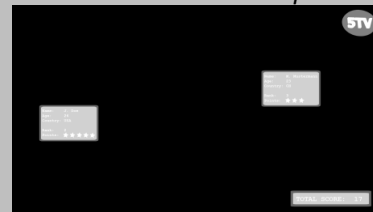
resSamples_{OL}



resSamples_{Label}



resSamples_{Alpha}



Data specified by user selection:

Optional Overlays

- program logo
- score bar
- information panel



user_label_cnt = 2
user_label_id[0] = 85
user_label_id[1] = 255
user_overlay_id[0] = 0





Summary and Recommendation

Summary

- Realization for Optional Overlay proposed with help of auxiliary picture layers
- Applications in sports broadcast, news and entertainment programs, advertisements, etc.
- Functionality requested by Disney and ESPN

Recommendation

- Adopt the Optional Overlays SEI message

