



# JCTVC-AF0024: Mask information for Annotated region SEI message

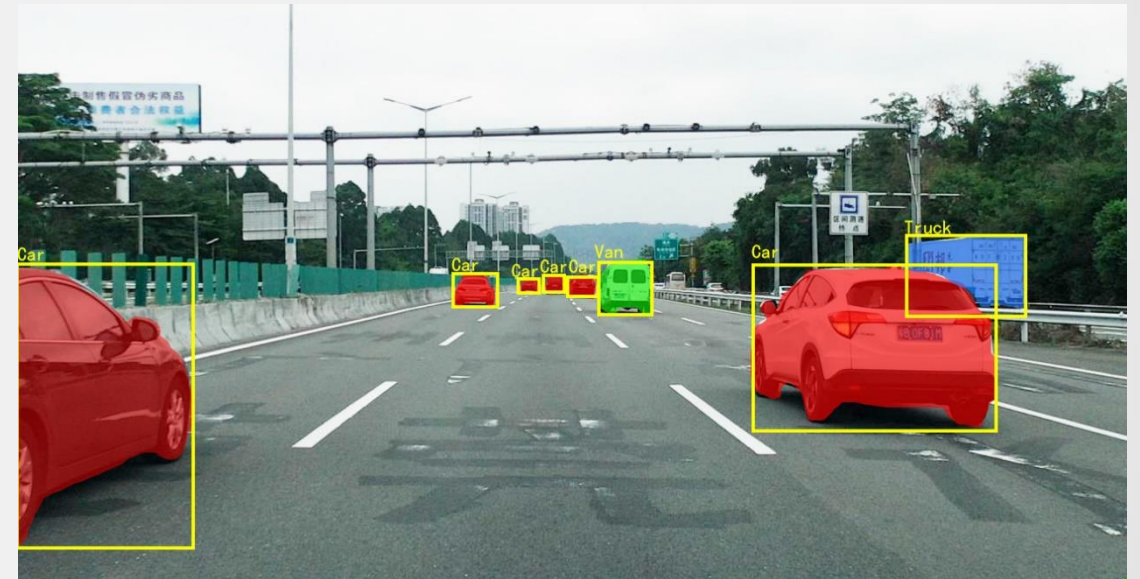
Xiaozhen Zheng, Xuyang Feng, Liliang Zhang, Cong Zhao

# Motivation

- Advantage of current Annotated Region SEI message
  - Avoid to perform same object tracking at encoder and decoder: Save computing power
    - Lots of devices have powerful computing capability at encoder side, e.g. drone
  - Get better tracking results from original video at encoder side
    - Decoder side can only perform analysis on compressed video

# Motivation (2/2)

- Shortcoming of Annotated Region SEI message: Can't describe shape of tracked objects
- Shape is useful for some applications
  - Example 1: highlight the tracked objects and dim background at surveillance scenario that help observer trace object easily
  - Example 2: Change background at decoder side interactive entertainment app



# Proposal: mask information for tracked objects

- One bit mask for tracked object: indicate a luma sample belongs to the object or not
- Mask information can be inferred from previous coded frame: Save signaling overhead
- A high level control flag to indicate whether mask is transmitted at current frame: Save signaling bits in some cases that don't need mask

# Modified syntax for Annotated SEI message

annotated_regions( payloadSize ) {	Descriptor
<b>ar_cancel_flag</b>	u(1)
<b>ar_not_optimized_for_viewing_flag</b>	u(1)
<b>ar_true_motion_flag</b>	u(1)
<b>ar_occluded_object_flag</b>	u(1)
<b>ar_partial_object_flag_present_flag</b>	u(1)
<b>ar_object_label_present_flag</b>	u(1)
<b>ar_object_confidence_present_flag</b>	u(1)
<b>ar_object_mask_present_flag</b>	<b>u(1)</b>
if( ar_object_confidence_present_flag )	
<b>ar_object_confidence_length_minus1</b>	u(4)
if( ar_object_label_present_flag ) {	
<b>ar_object_label_language_present_flag</b>	u(1)
if( ar_object_label_language_present_flag ) {	
while( !byte_aligned( ) )	
<b>ar_bit_equal_to_zero</b> /* equal to 0 */	f(1)
<b>ar_object_label_language</b>	st(v)
}	
<b>ar_num_cancelled_labels</b>	ue(v)
for( i = 0; i < ar_num_cancelled_labels; i++ )	
<b>ar_cancelled_label_idx[ i ]</b>	ue(v)
<b>ar_num_new_labels</b>	ue(v)

# Modified syntax for Annotated SEI message (2/3)

for( i = 0; i < ar_num_new_labels; i++ )	
<b>ar_label_idx[ i ]</b>	ue(v)
while( !byte_aligned( ) )	
<b>ar_bit_equal_to_zero</b> /* equal to 0 */	f(1)
<b>ar_label[ ar_label_idx[ i ] ]</b>	st(v)
}	
}	
<b>ar_num_cancelled_objects</b>	ue(v)
for( i = 0; i < ar_num_cancelled_objects; i++ )	
<b>ar_cancelled_object_idx[ i ]</b>	ue(v)
<b>ar_num_objects_minus1</b>	ue(v)
for( i = 0; i <= ar_num_objects_minus1; i++ ) {	
<b>ar_object_idx[ i ]</b>	ue(v)
<b>ar_new_object_flag[ ar_object_idx[ i ] ]</b>	u(1)
if( !ar_new_object_flag[ ar_object_idx[ i ] ] )	
<b>ar_bounding_box_update_flag[ ar_object_idx[ i ] ]</b>	u(1)
if ( ar_object_mask_present_flag ) {	
<b>ar_bounding_box_mask_present_flag[ ar_object_idx[ i ] ]</b>	u(1)
if ( ar_bounding_box_mask_present_flag[ ar_object_idx[ i ] ] && (!ar_new_object_flag[ ar_object_idx[ i ] ]    !ar_bounding_box_update_flag[ ar_object_idx[ i ] ] ) ) {	
<b>ar_bounding_box_mask_infer_flag[ ar_object_idx[ i ] ]</b>	u(1)
}	
}	

# Modified syntax for Annotated SEI message (3/3)

if( ar_new_object_flag[ ar_object_idx[ i ] && ar_object_label_present_flag )	
<b>ar_object_label_idc</b> [ ar_object_idx[ i ] ]	ue(v)
if( ar_partial_object_flag_present_flag )	
<b>ar_partial_object_flag</b> [ ar_object_idx[ i ] ]	u(1)
if( ar_object_bounding_box_update_flag[ ar_object_idx[ i ] ]    ar_new_object_flag[ ar_object_idx[ i ] ] ) {	
<b>ar_object_top</b> [ ar_object_idx[ i ] ]	u(16)
<b>ar_object_left</b> [ ar_object_idx[ i ] ]	u(16)
<b>ar_object_width</b> [ ar_object_idx[ i ] ]	u(16)
<b>ar_object_height</b> [ ar_object_idx[ i ] ]	u(16)
if( ar_object_confidence_present_flag )	
<b>ar_object_confidence</b> [ ar_object_idx[ i ] ]	u(v)
}	
if ( ar_bounding_box_mask_present_flag[ ar_object_idx[ i ] ] && !ar_bounding_box_mask_infer_flag[ ar_object_idx[ i ] ]    ar_new_object_flag[ ar_object_idx[ i ] ]    ar_bounding_box_update_flag[ ar_object_idx[ i ] ] ) {	
for (m = 0; m < ar_object_height[i]; m++)	
for (n = 0; n < ar_object_width[i]; n++)	
<b>mask</b> [ ar_object_idx[ i ] ][m][n]	u(1)
}	
}	
}	

# Example scenario: Picture 0

ar_object_label_present_flag	1	ar_object_idx[ 2 ]	2
ar_object_mask_present_flag	1	ar_new_object_flag[ 2 ]	1
ar_num_new_labels	4	ar_bounding_box_mask_present_flag[2]	1
ar_label_idx[ 0 ]	0	ar_object_idx[ 3 ]	3
ar_label_idx[ 1 ]	1	ar_new_object_flag[ 3 ]	1
ar_label_idx[ 2 ]	2	ar_bounding_box_mask_present_flag[3]	0
ar_label_idx[ 3 ]	3	ar_object_label_idc[ 0 ]	0
ar_label[ 0 ]	car	ar_object_label_idc[ 1 ]	1
ar_label[ 1 ]	car	ar_object_label_idc[ 2 ]	2
ar_label[ 2 ]	van	ar_object_label_idc[ 3 ]	3
ar_label[ 3 ]	truck	ar_object_top, left, width, height[ 0 ]	BB_A
ar_num_objects_minus1	3	ar_object_top, left, width, height[ 1 ]	BB_B
ar_object_idx[ 0 ]	0	mask[1]	car1's mask
ar_new_object_flag[ 0 ]	1	ar_object_top, left, width, height[ 2 ]	BB_C
ar_bounding_box_mask_present_flag[0]	0	mask[2]	van's mask
ar_object_idx[ 1 ]	1	ar_object_top, left, width, height[ 3 ]	BB_D
ar_new_object_flag[ 1 ]	1		
ar_bounding_box_mask_present_flag[1]	1		

At picture 0, 4 objects are present in the image, 2 car, 1 van and 1 truck.



# Example scenario: Picture 1

ar_object_label_present_flag	1	ar_new_object_flag[ 3 ]	1
ar_object_mask_present_flag	1	ar_object_bounding_box_update_flag[ 3 ]	0
ar_num_new_labels	0	ar_bounding_box_mask_present_flag[3]	0
ar_num_objects_minus1	3	ar_object_label_idc[ 0 ]	0
ar_object_idx[ 0 ]	0	ar_object_label_idc[ 1 ]	1
ar_new_object_flag[ 0 ]	0	ar_object_label_idc[ 2 ]	2
ar_object_bounding_box_update_flag[ 0 ]	0	ar_object_label_idc[ 3 ]	3
ar_bounding_box_mask_present_flag[0]	0	ar_object_top, left, width, height[ 0 ]	BB_A
ar_object_idx[ 1 ]	1	ar_object_top, left, width, height[ 1 ]	BB_B
ar_new_object_flag[ 1 ]	0	ar_object_top, left, width, height[ 2 ]	BB_E
ar_object_bounding_box_update_flag[ 1 ]	0	mask[2]	van's mask
ar_bounding_box_mask_present_flag[1]	1	ar_object_top, left, width, height[ 3 ]	BB_D
ar_bounding_box_mask_infer_flag[ 1 ]	1		
ar_object_idx[ 2 ]	2		
ar_new_object_flag[ 2 ]	0		
ar_object_bounding_box_update_flag[ 2 ]	1		
ar_bounding_box_mask_present_flag[2]	1		
ar_object_idx[ 3 ]	3		

At picture 1, the car0 (object 0), car1(object 1) and truck (object 3) stayed in the same position and van (object 2) moved to a new position. Car1's mask is inferred from previous frame

# Example scenario: Picture 2

ar_object_label_present_flag	1	ar_object_idx[ 3 ]	3
ar_object_mask_present_flag	1	ar_new_object_flag[ 3 ]	1
ar_num_new_labels	0	ar_object_bounding_box_update_flag[ 3 ]	0
ar_num_objects_minus1	3	ar_bounding_box_mask_present_flag[3]	0
ar_object_idx[ 0 ]	0	ar_object_label_idc[ 0 ]	0
ar_new_object_flag[ 0 ]	0	ar_object_label_idc[ 1 ]	1
ar_object_bounding_box_update_flag[ 0 ]	0	ar_object_label_idc[ 2 ]	2
ar_bounding_box_mask_present_flag[0]	0	ar_object_label_idc[ 3 ]	3
ar_object_idx[ 1 ]	1	ar_object_top, left, width, height[ 0 ]	BB_A
ar_new_object_flag[ 1 ]	0	ar_object_top, left, width, height[ 1 ]	BB_B
ar_object_bounding_box_update_flag[ 1 ]	0	mask[1]	car1's mask
ar_bounding_box_mask_present_flag[1]	1	ar_object_top, left, width, height[ 2 ]	BB_E
ar_bounding_box_mask_infer_flag[ 1 ]	0	ar_object_top, left, width, height[ 3 ]	BB_D
ar_object_idx[ 2 ]	2		
ar_new_object_flag[ 2 ]	0		
ar_object_bounding_box_update_flag[ 2 ]	0		
ar_bounding_box_mask_present_flag[2]	1		
ar_bounding_box_mask_infer_flag[ 2 ]	1		

At picture 2, all of objects stayed in the same position. Car1's mask is updated and van's mask is inferred from previous frame.

# Summary

- Propose mask information based on Annotated Region SEI message
- Enable signal mask for tracked objects with high efficiency and flexibility
- Be useful at multiple decoder side applications, e.g. surveillance, interactive entertainment



THANK YOU