

JVET-AE0280 AHG9: Common text for proposed generative face video SEI message

B. Chen, J. Chen, Y. Ye (Alibaba)

S. Wang (CityU)

S. McCarthy, P. Yin, G.-M. Su, A. K. Choudhury, W. Husak (Dolby)

Introduction

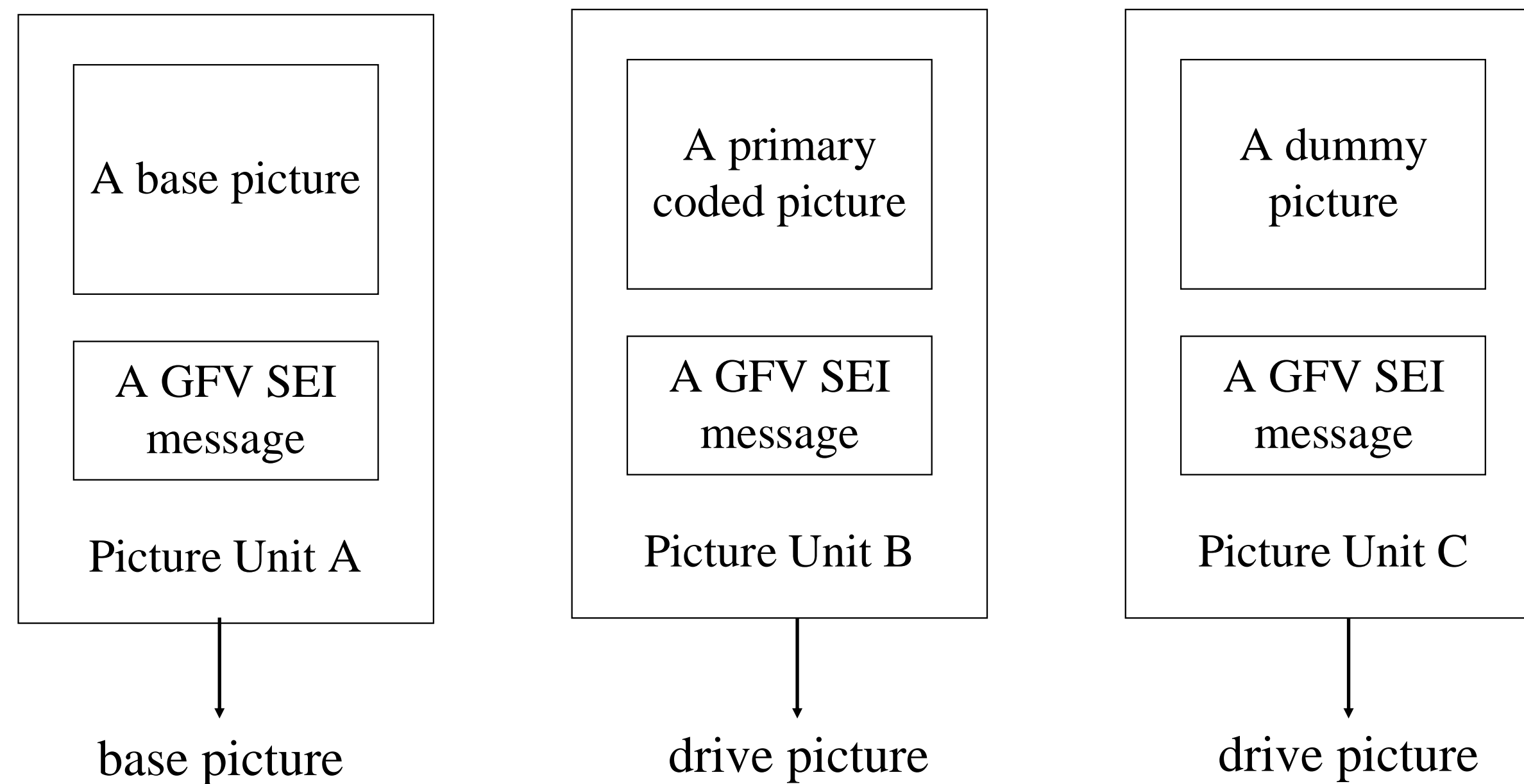
- Generative Face Video (GFV) SEI message were proposed in JVET-AC0088 and JVET-AD0051 at the 29th and 30th JVET meetings, respectively.
- In this meeting, JVET-AE0080 and JVET-AE0083 were submitted to address the comments noted during the 30th JVET meeting.
- This contribution proposes common text based on JVET-AE0080 and JVET-AE0083.
 - refined Syntax and semantics
 - interface with generative model

Proposal

- The proposed GFV SEI message syntax supports the current mainstream facial feature representation methods and is generalized to cover other potential methods which may be used in the future.
 - 2D/3D Landmarks
 - 2D keypoints + affine transformation matrix
 - Region matrix
 - 3D keypoints
 - Compact feature matrix
 - Facial semantics
 -

Proposal

- Each picture unit (PU) may contain a GFV SEI message.



- ✓ a base picture: a decoded output picture that may be used by a network to generate a novel face picture
- ✓ a dummy picture: a picture unit of the minimum allowed resolution that contains only SEI messages
- ✓ a primary coded picture: a picture can be fused by a generative network to improve background texture and facial details

- To solve “encoder-decoder match”:

- Signal model with ISO/IEC 15938-17
- Identify the model with URI
- Signal a key to identify whether the analysis network at the encoder matches with the generative network at the decoder

Proposal

- gfv_base_pic_flag**: whether the current picture is base picture
- gfv_drive_pic_fusion_flag**: whether the current derive picture is input to generative model
- gfv_id**: to identify face feature information and generative model specified by this SEI message
- gfv_key**: to determine whether the analysis network at the encoder matches with the generative network at the decoder

generative_face_video (payloadSize) {	Descriptor
gfv_id	ue(v)
gfv_base_pic_flag	u(1)
if(gfv_base_pic_flag) { /*specify Generator()*/	
gfv_key_present_flag	
if(gfv_key_present_flag) {	
gfv_key	u(32)
else {	
gfv_nn_base_flag	u(1)
gfv_nn_mode_idc	ue(v)
if(gfv_nn_mode_idc == 1) {	
while(!byte_aligned())	
gfv_nn_reserved_zero_bit_a	u(1)
gfv_nn_tag_uri	st(v)
gfv_nn_uri	st(v)
}	
}	
} else	
gfv_drive_pic_fusion_flag	ue(v)
...	
....	
if(gfv_base_pic_flag && !gfv_key_present_flag)	
if(gfv_nn_mode_idc == 0) {	
while(!byte_aligned())	
gfv_nn_reserved_zero_bit_b	u(1)
for(i = 0; more_data_in_payload(); i++)	
gfv_nn_payload_byte[i]	b(8)
}	
}	

Proposal

- **gfv_coordinate_present_flag**: whether keypoints are present
- **gfv_matrix_present_flag**: whether matrice are present
- **gfv_kp_pred_flag**: whether directly signal the coordinates or predictively signal the coordinates
- **gfv_coordinate_z_present_flag**: is 2D keypoint or 3D keypoint
- **gfv_matrix_type_idx**: matrix type specified in below table

Value	Specification	Size
0	Affine translation matrix	2*2 or 3*3
1	Covariance matrix	2*2 or 3*3
2	Mouth matrix representing mouth motion.	Specified by SEI message
3	Eye matrix representing the open-close status and level of eyes.	Specified by SEI message
4	Head rotation parameters representing the head rotation.	2*2 or 3*3
5	Head translation matrix representing head translation	1*2 or 1*3
6	Head location matrix with size of representing the head location	1*2 or 1*3
7	Compact feature matrix	Specified by SEI message
8...31	Other matrix that may be used as determined by the application	Specified by SEI message
32...63	Reserved	

....	
gfv_coordinate_present_flag	u(1)
if(gfv_coordinate_present_flag) {	
gfv_coordinate_precision_factor_minus1	ue(v)
gfv_num_kp_minus1	ue(v)
gfv_kp_pred_flag	u(1)
gfv_coordinate_z_present_flag	u(1)
if(gfv_coordinate_z_present_flag)	
gfv_coordinate_z_max_value_minus1	ue(v)
for(i = 0; i <= num_kp_minus1; i++) {	
if(!gfv_kp_pred_flag) {	
//signal coordinate value for each keypoint	u(v)
}	
else	u(1)
//signal coordinate delta value for each keypoint	u(v)
}	
gfv_matrix_present_flag	u(1)
if(gfv_matrix_present_flag) {	
gfv_matrix_element_precision_factor_minus1	ue(v)
gfv_num_matrix_types_minus1	ue(v)
for(i = 0; i <= num_matrix_types_minus1; i++) {	
gfv_matrix_type_idx [i]	u(6)
//signal matrix number, width, height information dependent on matrix type	
// signal each element value of each matrix	

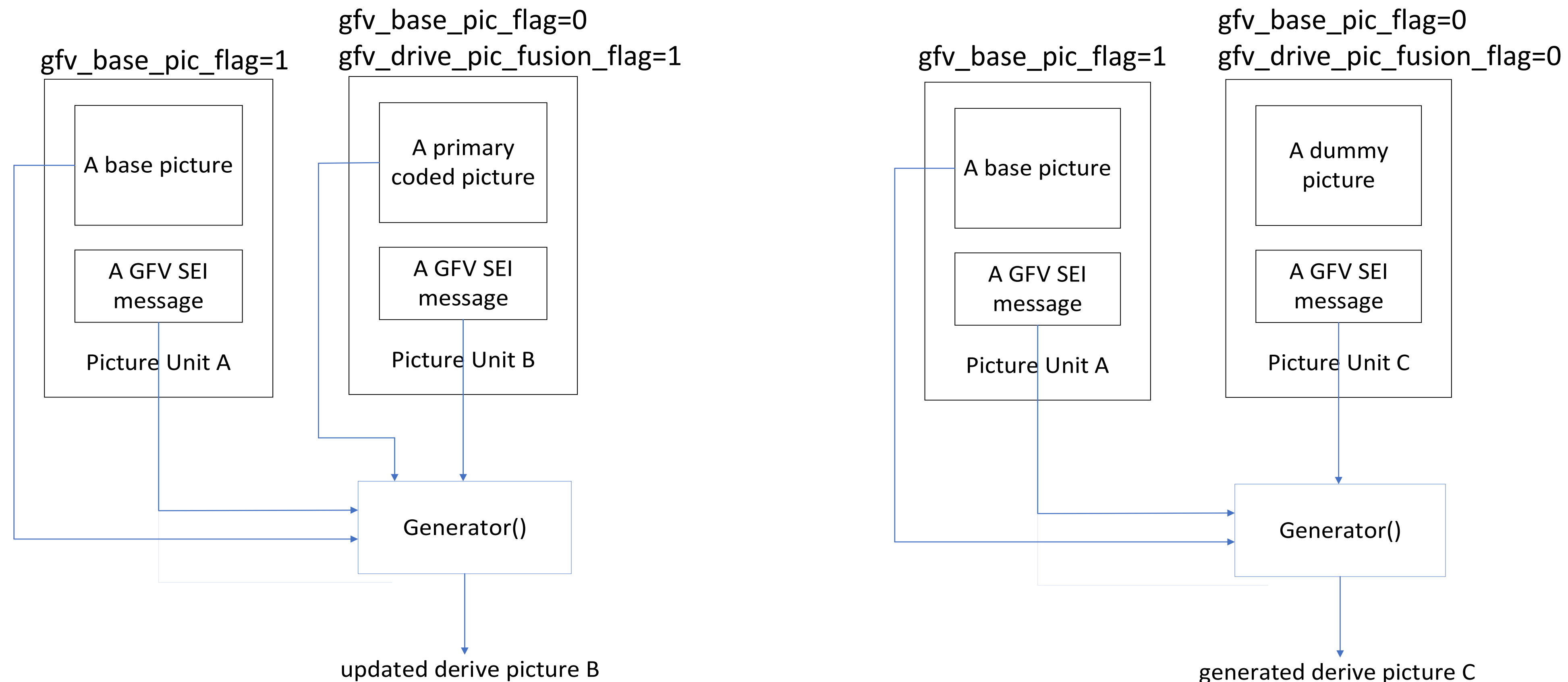
Proposal

- Examples of syntax configuration for the current main-stream generative models

Generative model	Facial representation	Coordinate syntax	Matrix syntax
VSBNet	2D Landmarks	98 2D coordinates	N/A
FOMM	2D Keypoints + Affine Matrix	10 2D coordinates	10 2x2 matrices (affine)
MRAA	2D Keypoints + Affine Matrix+ Covariance Matrix	10 2D coordinates	20 2*2 matrices (10 are affine and 10 are covariance)
Face_vid2vid	3D Keypoints + Rotation Matrix+ Translation Matrix	15 3D coordinates	1 3*3 matrix (rotation) 1 1*3 matrix (translation)
CFTE	Compact Feature Matrix	N/A	1 4*4 matrix (compact feature) 1 1*6 matrix (mouth)
IFVC	Mouth Matrix + Eye Matrix + Rotation Matrix+ Translation Matrix + Location Matrix	N/A	1 1*1 matrix (eye) 1 1*3 matrix (rotation) 1 1*3 matrix (translation) 1 1*1 matrix (location)

Proposal

- Denote the generative model to generate the output picture as **Generator()**
- Define the **inputs and outputs** of Generator()
- Define **the process to derive the inputs** of Generator() based on the syntax elements in GFV SEI message and **convert the outputs** of Generator() to picture sample array



Summary

- The proposed generative face video SEI
 - achieves ultra-low bitrate compression at a fraction of VVC's bitrate using powerful generative networks.
 - enables encoder-guided face animation and filtering functionality on many popular mobile apps.
- To address comments previously received, the common text provided in this contribution
 - refines the syntax and semantics.
 - provides interface definition between decoder and the generative model.
- Suggest to adopt the proposed SEI message to VSEI V4 working draft.

Thanks