



# JCTVC-F291: Picture Orientation Information

Danny Hong, Jill Boyce, Stephan Wegner

Vidyo



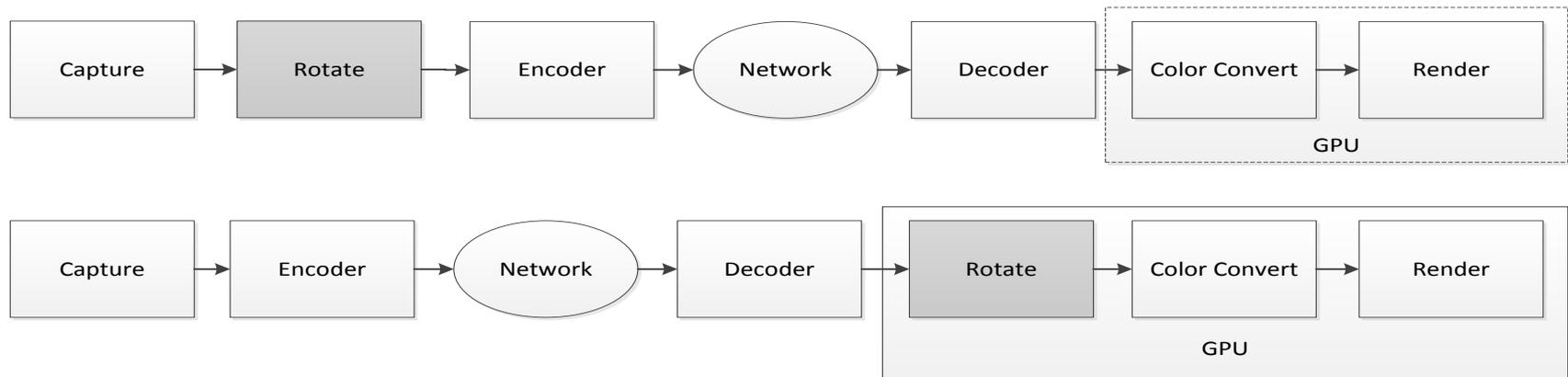
# JCTVC-F291: Picture Orientation Information



- JCTVC-E280 proposed allowing the coded picture orientation to change on a per sequence basis
  - Signaled in the VUI, SPS, or SEI message
- This contribution proposes an extension to allow orientation to change on a picture-by-picture basis, without requiring an IDR picture
  - Signaled in the PPS or slice header

# Motivation

- ◆ Some video capturing devices can capture pictures in orientation different from the rendering/display orientation
  - Orientation may change at any time
- ◆ Rotation can be performed at the capturing end, prior to encoding, or at the display end, after decoding
  - Can take advantage of GPUs
  - Small coding efficiency gains in some cases



# Proposal

- ◆ Add orientation information per picture
  - PPS or slice header
  - 4 rotation positions – 0, 90, 180, or 270 degrees

<code>pic_parameter_set_rbsp( ) {</code>	Descriptor
<code>...</code>	
<code>orientation_idc</code>	<code>u(2)</code>
<code>...</code>	
<code>}</code>	

**orientation\_idc** \* 90 degrees specifies the clockwise rotation of the decoded picture need for proper rendering. `orientation_idc` shall be in the range of 0 to 3, inclusive.

# Proposal

- **JCTVC-E280 did not require any change to the encoding or decoding process, and merely used post-processing**
- **This contribution requires change to encoding and decoding process to rotate the reference picture and temporal motion vector predictor, if reference picture orientation differs from current picture orientation**
- **The experimental results in JCTVC-E280 used naïve encoder, encoded each sequence at 0 and 90 degrees, for all frames in sequence**
- **This contribution proposes non-normative per-frame decision processes**
  - Full RDO decision
  - Fast method (for Intra only)

# Experimental Conditions

- HM 3.0 is modified to support  $\text{orientation\_idc} = 0$  and 1
- Full RDO method
  - Each picture is encoded twice; first with  $\text{orientation\_idc} = 0$  and then with  $\text{orientation\_idc} = 1$
  - The orientation that yields better RD cost is chosen
- Fast method (for intra only)
  - Each picture is downsampled to 1/16 of original resolution
  - Sobel operator is applied to the downsampled picture to derive the direction of details
  - If there are more forward diagonal details than backward diagonal details, then  $\text{orientation\_idc} = 1$  is chosen

# Experimental Results (Full RDO) LoCo



	Intra			Random Access			Low Delay		
	Y BD-rate	U BD-rate	V BD-rate	Y BD-rate	U BD-rate	V BD-rate	Y BD-rate	U BD-rate	V BD-rate
Class A	-1.7	-1.3	-1.8	-1.5	0.1	-1.1			
Class B	-0.6	-1.5	-2.0	-0.4	-1.1	-2.0	0.0	0.1	-0.3
Class C	-0.1	0.1	0.0	-0.2	0.1	0.1	-0.1	0.3	0.1
Class D	-0.3	-0.2	0.0	-0.2	-0.6	-0.1	0.0	-0.3	0.0
Class E	-0.3	-2.0	-1.7				-0.3	-1.2	-1.9
<b>All</b>	<b>-0.6</b>	<b>-0.9</b>	<b>-1.1</b>	<b>-0.6</b>	<b>-0.4</b>	<b>-0.8</b>	<b>-0.1</b>	<b>-0.2</b>	<b>-0.4</b>
<b>Max</b>	<b>-3.4</b>	<b>-3.2</b>	<b>-4.4</b>	<b>-2.8</b>	<b>-2.1</b>	<b>-3.9</b>	<b>-0.3</b>	<b>-2.1</b>	<b>-2.5</b>

# Experimental Results (Fast) LoCo

	Intra		
	Y BD-rate	U BD-rate	V BD-rate
Class A	-1.6	-1.1	-1.4
Class B	-0.6	-1.5	-2.0
Class C	0.0	0.0	0.0
Class D	-0.2	-0.1	0.0
Class E	0.1	-0.4	-0.1
<b>All</b>	<b>-0.5</b>	<b>-0.7</b>	<b>-0.8</b>
<b>Max</b>	<b>-3.4</b>	<b>-3.2</b>	<b>-4.4</b>

# Conclusions

- ◆ Include orientation\_idc in the PPS or the slice header to support difference picture orientations on a picture-by-picture basis, without requiring an IDR picture
- ◆ Benefits:
  - ◆ The encoding side can encode each picture in the captured orientation, deferring the picture rotation process to the decoder
  - ◆ Coding gain may be realized