

REDEFINING MOBILITY



# JCT3V-E0133: Inter-view motion vector prediction for depth coding

Vijayaraghavan Thirumalai, Li Zhang, Ying Chen, Marta Karczewicz (Qualcomm)

# Summary

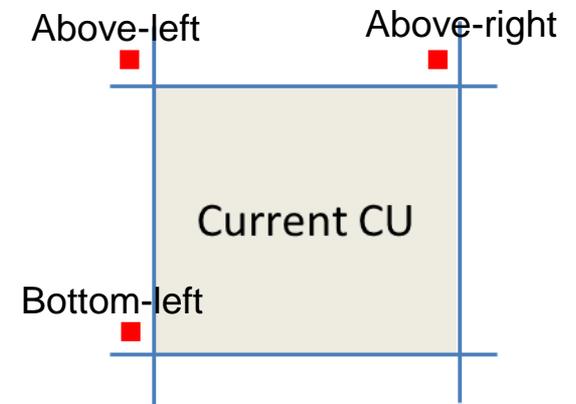
- Proposed to extend the inter-view motion prediction scheme to depth coding
  - A disparity vector is calculated from the neighboring samples.
  - Inter-view motion candidates are derived and inserted in the merge list.
  - A more accurate reference block selection method is applied for deriving the inter-view motion candidates.
  
- Proposed method provides an improved coding gain of 0.2%

# Introduction

- Merge candidate list construction in 3D-HEVC for depth coding
  1. MPI candidate
  2. Spatial merging candidates
  3. Temporal merging candidate
  4. Other merging candidates
    - Combined bi-predictive merging candidates
    - Zero motion vector merging candidates

# Disparity vector calculation

- Disparity vector (DV) is necessary to derive the inter-view motion prediction candidates
- DV is calculated for each CU from a single depth value
  - A depth value is derived from three neighboring samples



$$\text{Depth} = ( 5 * \underbrace{D[xC-1][yC-1]}_{\text{depth at above-left}} + 5 * \underbrace{D[xC-1][yC+2N-1]}_{\text{depth at bottom-left}} + 6 * \underbrace{D[xC+2N-1][yC-1]}_{\text{depth at above-right}} + 8 ) \ggg 4$$

# Generation of inter-view motion candidates

- Two candidates are derived based on the derived disparity vector
  1. Inter-view motion vector candidate (IPMC)
  2. Inter-view disparity motion vector candidate
- More accurate IPMC is generated from a reference block identified by shifting the center pixel by (1,1)
  - Reference pixel position in the current HTM is derived as

$$\begin{aligned}xRef &= \text{Clip3}( 0, \text{PicWidthInSamples}_L - 1, xP + ( (nPSW-1) \gg 1 ) + ( (DV[ 0 ] + 2 ) \gg 2 ) ) \\yRef &= \text{Clip3}( 0, \text{PicHeightInSamples}_L - 1, yP + ( (nPSH-1) \gg 1 ) + ( (DV[ 1 ] + 2 ) \gg 2 ) )\end{aligned}$$

- Proposed method to derive the reference pixel position

$$\begin{aligned}xRef &= \text{Clip3}( 0, \text{PicWidthInSamples}_L - 1, xP + ( (nPSW-4) \gg 1 ) + ( (DV[ 0 ] + 2 ) \gg 2 ) ) \\yRef &= \text{Clip3}( 0, \text{PicHeightInSamples}_L - 1, yP + ( (nPSH-4) \gg 1 ) + ( (DV[ 1 ] + 2 ) \gg 2 ) )\end{aligned}$$

# Proposed merge candidate list

- Newly inserted candidates are marked in red
  1. MPI candidate
  2. **IPMC candidate is inserted only if it is different from the MPI candidate**
  3. Spatial merging candidates  $A_1$ ,  $B_1$  and  $B_0$
  4. **Inter-view disparity motion vector candidate is inserted only if it is different from the spatial merging candidates  $A_1$  and  $B_1$**
  5. Spatial merging candidate  $A_0$  and  $B_2$
  6. Temporal merging candidates
  7. Other merging candidates
    - Combined bi-predictive merging candidates
    - Zero motion vector merging candidates

# Experimental results

- Platform: HTM7.0r1
- Test conditions: CTC

Table 1: Results of the proposed method compared to anchor

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time	ren time
Balloons	0.0%	-0.4%	-0.5%	-0.2%	-0.2%	-0.3%	95.5%	104.4%	87.0%
Kendo	0.0%	-0.2%	-0.4%	-0.1%	-0.1%	-0.4%	101.1%	105.0%	92.0%
Newspaper_CC	0.0%	-0.2%	-0.2%	-0.1%	-0.1%	0.3%	96.3%	101.5%	88.4%
GT_Fly	0.0%	0.1%	-0.1%	0.0%	0.0%	-0.2%	94.6%	98.2%	87.9%
Poznan_Hall2	0.0%	-0.6%	-0.5%	-0.2%	-0.1%	-0.3%	104.8%	106.2%	96.4%
Poznan_Street	0.0%	0.1%	0.2%	0.0%	0.0%	0.0%	95.4%	101.1%	91.4%
Undo_Dancer	0.0%	-0.4%	-0.2%	-0.1%	-0.1%	-0.3%	98.1%	98.8%	90.6%
1024x768	0.0%	-0.3%	-0.4%	-0.1%	-0.1%	-0.2%	97.6%	103.6%	89.1%
1920x1088	0.0%	-0.2%	-0.2%	-0.1%	0.0%	-0.2%	98.2%	101.1%	91.6%
<b>average</b>	<b>0.0%</b>	<b>-0.2%</b>	<b>-0.2%</b>	<b>-0.1%</b>	<b>-0.1%</b>	<b>-0.2%</b>	<b>98.0%</b>	<b>102.2%</b>	<b>90.5%</b>

# Experimental results

- Performance of our scheme on top of proposals JCT3V-E0156 and JCT3V-E0157

Table 2: Results of the proposed method integrated on top of SDC-based coding methods

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time	ren time
Balloons	0.0%	-0.6%	-0.5%	-0.2%	-0.4%	-1.0%	107.7%	107.2%	89.0%
Kendo	0.0%	0.0%	-0.4%	-0.1%	-0.5%	-1.5%	108.6%	105.8%	89.1%
Newspaper_CC	0.0%	-0.2%	-0.1%	-0.1%	-0.4%	-1.6%	109.8%	103.0%	89.5%
GT_Fly	0.0%	-1.4%	-1.2%	-0.3%	0.0%	-1.3%	101.4%	104.8%	91.1%
Poznan_Hall2	0.0%	-0.5%	0.1%	-0.1%	0.2%	-1.7%	106.0%	100.6%	91.2%
Poznan_Street	0.0%	-0.2%	0.2%	0.0%	0.1%	-0.3%	98.8%	97.1%	89.5%
Undo_Dancer	0.0%	-0.9%	-0.6%	-0.2%	0.1%	-1.2%	99.2%	101.1%	89.5%
1024x768	0.0%	-0.3%	-0.3%	-0.1%	-0.4%	-1.3%	108.7%	105.3%	89.2%
1920x1088	0.0%	-0.7%	-0.4%	-0.2%	0.1%	-1.1%	101.4%	100.9%	90.3%
<b>average</b>	<b>0.0%</b>	<b>-0.5%</b>	<b>-0.4%</b>	<b>-0.1%</b>	<b>-0.1%</b>	<b>-1.2%</b>	<b>104.5%</b>	<b>102.8%</b>	<b>89.8%</b>

# Experimental results

- Performance of our scheme when MPI is disabled.

Table 3: Results of the proposed method compared to anchor (both with MPI disabled)

	video 0	video 1	video 2	video PSNR / video bitrate	video PSNR / total bitrate	synth PSNR / total bitrate	enc time	dec time	ren time
Balloons	0.0%	-0.5%	-0.4%	-0.2%	-1.2%	-1.7%	101.0%	100.5%	93.6%
Kendo	0.0%	-0.2%	-0.4%	-0.1%	-1.2%	-2.1%	94.7%	103.6%	88.5%
Newspaper_CC	0.0%	-0.1%	-0.1%	-0.1%	-0.8%	-0.4%	95.1%	103.6%	89.1%
GT_Fly	0.0%	-0.4%	0.0%	-0.1%	-0.1%	-0.7%	107.6%	109.5%	102.0%
Poznan_Hall2	0.0%	-0.3%	-0.3%	-0.1%	-1.8%	-3.1%	98.4%	105.2%	90.0%
Poznan_Street	0.0%	0.0%	0.3%	0.0%	-0.7%	-1.2%	94.9%	98.9%	88.9%
Undo_Dancer	0.0%	-0.4%	-0.2%	-0.1%	-0.8%	-1.4%	94.2%	102.2%	88.0%
1024x768	0.0%	-0.3%	-0.3%	-0.1%	-1.1%	-1.4%	96.9%	102.5%	90.4%
1920x1088	0.0%	-0.3%	-0.1%	-0.1%	-0.8%	-1.6%	98.8%	104.0%	92.2%
<b>average</b>	<b>0.0%</b>	<b>-0.3%</b>	<b>-0.2%</b>	<b>-0.1%</b>	<b>-1.0%</b>	<b>-1.5%</b>	<b>98.0%</b>	<b>103.4%</b>	<b>91.4%</b>

- Thanks to the LG for the cross-check (JCT3V-E0202)

# Conclusions

- The proposed scheme is a simple extension of texture inter-view motion prediction to depth coding
  - More merge candidates are derived from the motion information in the reference block pointed by the disparity vector.
- Main results
  - The coding efficiency is improved by 0.2%
  - The coding gain is shown to be fully additive with the combined proposals JCT3V-E0156 and JCT3V-E0157

# Thank you!