QUALCOMM CDMA Technologies

REDEFINING MOBILITY

JCT3V-C0049: CE4: Advanced residual prediction for multiview coding

Li Zhang, Ying Chen, Xiang Li and Marta Karczewicz

Summary

- A follow-up of JCT3V-C0049 with the following modification
  - Removal of the parsing dependence of weighting factor index on the reference picture types

- Further simplifications of ARP
  - Simp #1: reference pictures are fixed for all PUs coded with residual prediction
  - Simp #2: in addition to Simp #1, bi-linear filter instead of the 8/4-tap luma/chroam interpolation filter is used when generating the residual of current PU.

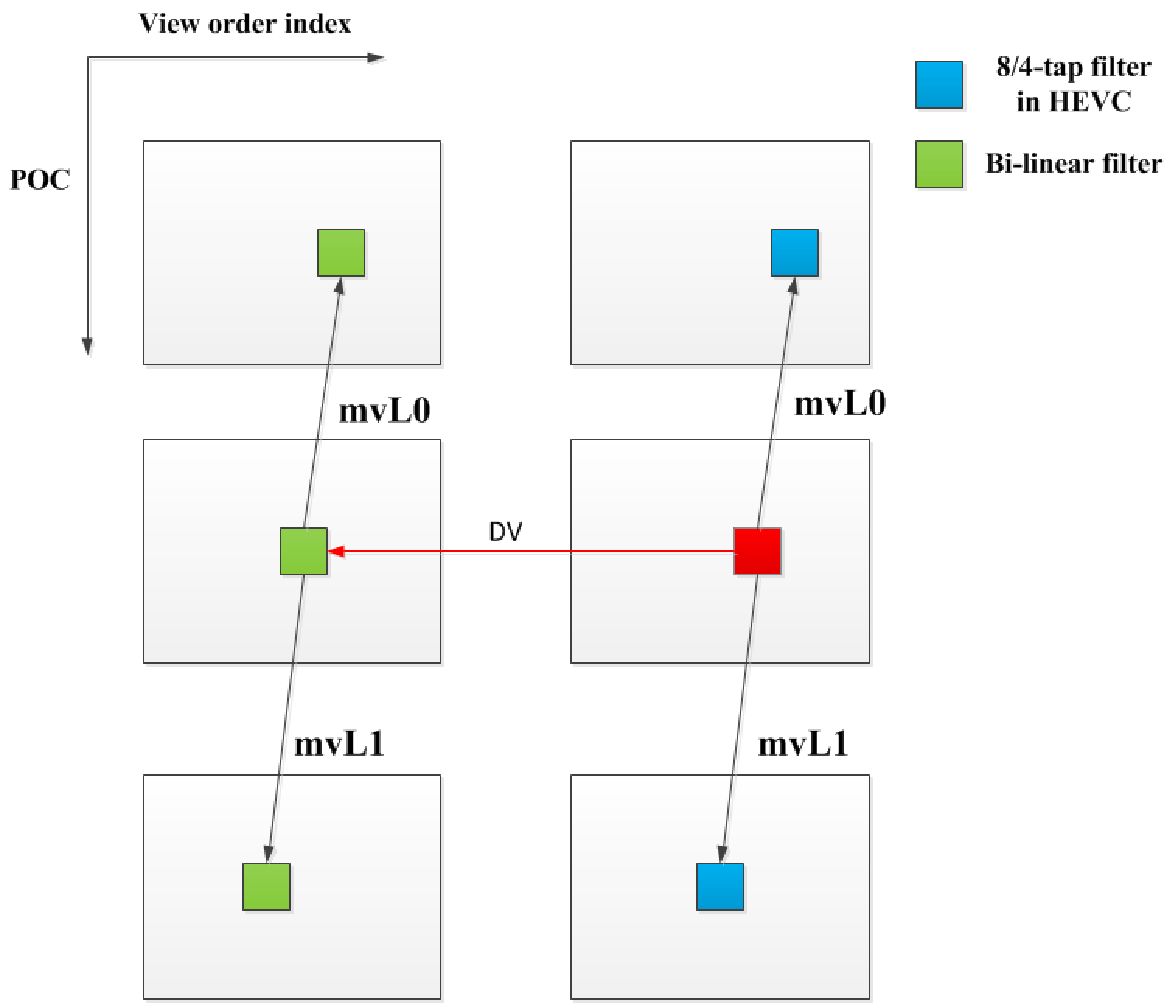
- Simulation results report significant coding gain and decoding time saving.

Background

- Inter-view residual prediction (IVRP) for dependent texture views in 3D-HEVC
  - IVRP is enabled for the inter-view merging candidate with PU size equal to 2Nx2N
  - Procedures
    - Locate a reference block in the base view based on the disparity vector.
    - Generate the residue using bi-linear filter of the reference block and use it as a predictor of the residue of current PU.
    - Transmit the difference of current residual and the residual predictor.
- Problems
  - Reconstructed residues are used which introduces undesired quantization error in prediction and further degrades the performance.
  - IVRP is only not applied to other merging/AVMP candidates except the inter-view merging candidates. The prohibition of the usage of IVRP restricts the coding gain of IVRP. Meanwhile, when inter-view motion prediction is disabled, IVRP no longer has any effect.
  - One additional image shall be allocated to store the residual information of the reference view.

Proposed Method (as JCT3V-C0049)

- Prediction structure of ARP

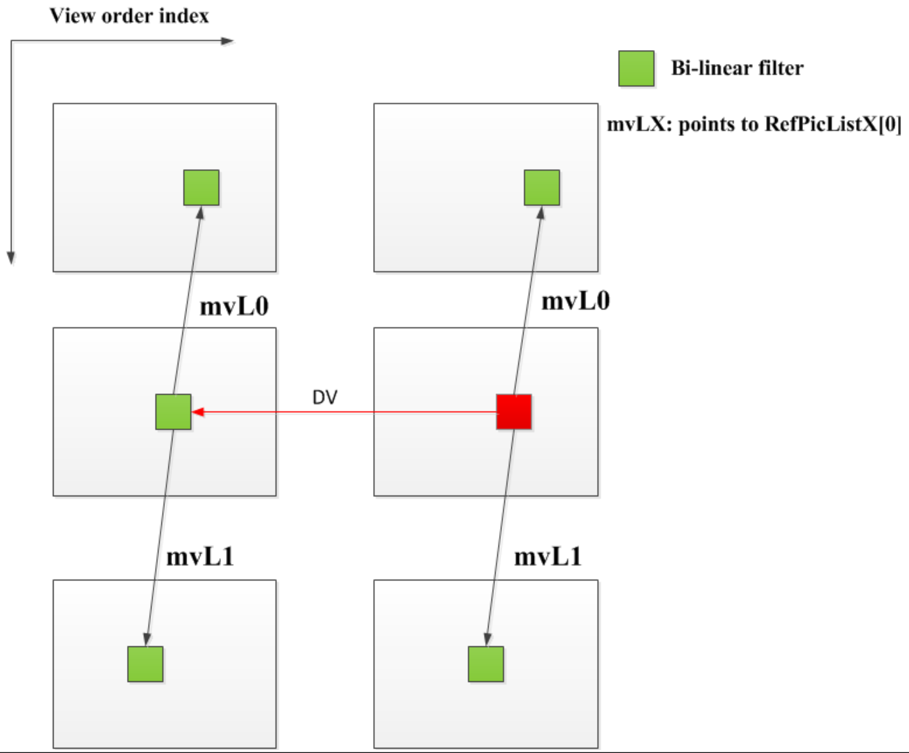


Proposed Method (as JCT3V-C0049)

- Motion is aligned
  - Reuse the motion information of current CU to the reference block
  - Apply motion compensation to the reference block to derive the residual block
- Interpolation during the residual generation process (non-base view):
  - 8-tap and 4-tap interpolation filters in HEVC
- Interpolation during the residual predictor generation process (base (reference) view):
  - Bi-linear filter
- Weighting factor unequal to 0 or 1 could be applied in ARP
  - Three factors are supported: 0, 0.5 and 1
    - Signaled in CU level with partition mode equal to 2Nx2N
  - Residual block is weighted with the selected weighting factor

Further Simplifications

- Simplification #1
  - Fixed reference pictures for all blocks coded with ARP
    - – The first entry of the reference picture list is selected and MV is scaled
- Simplification #2
  - Combined with simplification #1, bi-linear filter instead of 8/4-tap filters for non-base texture views is applied when residual prediction is used.



Simulation Results

- Coding performance of IVRP and ARP
  - HTM v6.0
  - Anchor: CTC with IVRP disabled

Table 1: Coding performance of IVRP and ARP

Coding tools	Simulation results					
	Video 1	Video 2	Video PSNR / Video bitrate	Video PSNR / total bitrate	Synth PSNR / total bitrate	Dec Time
IVRP	**-1.2%**	**-1.3%**	**-0.5%**	**-0.5%**	**-0.4%**	**112%**
ARP w/o simplifications	**-3.7%**	**-3.7%**	**-1.3%**	**-1.2%**	**-1.1%**	**101%**
Simp #1	**-3.5%**	**-3.4%**	**-1.3%**	**-1.2%**	**-1.0%**	**102%**
Simp #2	**-3.3%**	**-3.2%**	**-1.1%**	**-1.0%**	**-0.9%**	**99%**

Simulation Results

- Coding performance of ARP
  - HTM v6.0
  - Anchor: CTC with IVRP on

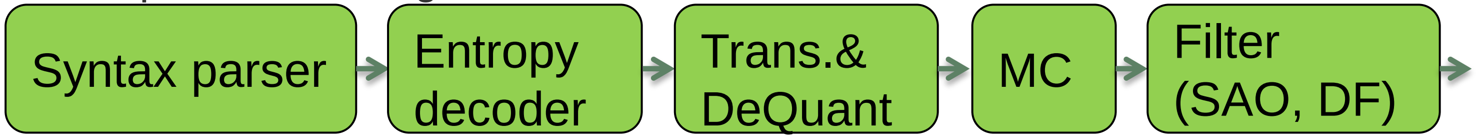
Table 2: Coding performance of ARP

Coding tools	Simulation results					
	Video 1	Video 2	Video PSNR / Video bitrate	Video PSNR / total bitrate	Synth PSNR / total bitrate	Dec Time
ARP w/o simplifications	**-2.5%**	**-2.4%**	**-0.8%**	**-0.8%**	**-0.6%**	**90%**
Simp #1	**-2.3%**	**-2.1%**	**-0.7%**	**-0.7%**	**-0.6%**	**90%**
Simp #2	**-2.1%**	**-1.9%**	**-0.6%**	**-0.6%**	**-0.5%**	**88%**

- Thanks to LG for the cross-check (JCT3V-D0093)

Hardware implementation implications

- Number of cycles
  - multiple pipeline stages for decoders
  - MC: 60%-80% of cycle budget
    - ➢20%-40% of cycle budget are spare, enough for ARP
    - ➢a parallel MC engine with reasonable hardware cost



- Increase of hardware cost (silicon area size)
  - MC engine takes about 10% or a little less logic on chip
  - The increased design logic of MC in ARP could be much less than 10% since bi-linear filter is applied instead of 8/4-tap filters (the same as IVRP).
- Power consumption
  - For ARP, the power increase is roughly the same as the design logic increase.
  - More intelligently power management scheme, e.g., clock gating, may apply to significantly reduce the average power increase, to the same level as existing HEVC design since ARP is not always selected.

Hardware implementation implications

- Storage and memory bandwidth
  - DPB memory
    - ARP: no memory increase, i.e., decoded pictures present in the DPB are utilized.
    - IVRP: needs to generate and store a decoded residual picture for the base view.
  - Memory access inside the L2 cache (on-chip memory)
    - Possible reference regions may be loaded into the L2 cache in advance to avoid real-time data fetching beyond L2 cache and the fetching of external memory into the L2 cache can be done in the background pipeline much earlier to avoid on-the-fly fetching of external memory.
    - A trade-off between memory size of the L2 cache and the chances of cache misses, which require on-the-fly fetching external memory from DPB.
    - The fact of possibly more memory required in L2 may be accommodated by two factors:
      - More advanced mechanism to manage which reference picture regions to be pre-fed into the L2 cache;
      - Upgraded high-end SoC equipped with more memory in the near future.
    - If the potential increase of L2 cache to reduce cache misses in ARP is considered as a serious issue, the simplified methods, i.e., unifying to just one frame per list by scaling the MV could be used.

Hardware implementation implications

- Storage and memory bandwidth
  - Memory access from external memory ((bi-prediction, 8x8)

Coding tools	**Num. of pixels to be accessed**			
	Non-base view	Base view (reference view)	Num. of accessed pixels for one pixel	Ratio to HEVC
HEVC	(8+7)*(8+7)*2 **(= 450)**	NA	450/64 = **7.0**	NA
IVRP	**450**	(8+1)*(8+1) (= **81**)	(450+81)/64 = **8.3**	**118%**
ARP w/o simplifications	**450**	((8+1)*(8+1)*3) (= **243**)	(450+243)/64 = **10.8**	**154%**
Simp #1	**450**	**243**	**10.8**	**154%**
Simp #2	(8+1)*(8+1)*2 (= **162**)	**243**	(162+243)/64 = **6.3**	**90%**

Complexity Analysis

- For the worst case, the complexity of IVRP, ARP and its simplified versions is compared to HEVC single view coding
  - Use the template JCTVC-L0440*

Table 3: Analysis of complexity of IVRP and ARP for the worst case (based on 4x2 pattern)

Coding tools	Num. Mul	Num. Add	Memory Bandwidth
IVRP	109%	106%	123%
ARP	128%	117%	169%
ARP simp #1	128%	117%	169%
ARP simp #2	47%	28%	115%

*E. François, A. Tabatabai, E. Alshina, BoG report: Methodology for evaluating complexity of combined and residual prediction methods in SHVC, JCTVC-L0440, Geneva, Switzerland, 14–23 Jan. 2013.

Conclusion

- ARP achieves much higher coding gain while the complexity increase is limited or even less than current IVRP with simp #2.

Table 4: Summarization of comparisons of IVRP and ARP

Coding tools	Coding gain (vs HTM w/oIVRP)	Potential memory increase in L2 cache	Memory Bandwidth	Num. Mul	Num. Add
IVRP	0.5%	1 (O(1))	123%	109%	106%
ARP	1.3%	N (O(n))	169%	128%	117%
ARP simp #1	1.3%	2 (O(1))	169%	128%	117%
ARP simp #2	1.1%	2 (O(1))	115%	47%	28%

Thank you!