

Description of SDR, HDR, and 360° video coding technology proposal by InterDigital and Dolby

JVET-J0015

Introduction

- JEM is an experimental platform
 - Provides good coding performance
 - For CS1, JEM anchors provide 33.15% average BD rate savings over HM CfP anchors
 - But has many complexity issues, not just in terms of encoding/decoding time
- Many JEM coding tools are well understood by the experts, and some simplified flavor of them will likely be included in our next standard
- We target simplification with the following important design considerations for the future video coding standard

Decoder complexity is still one of the most important design aspects

Any increase in decoder complexity must be justified

Real-time decoder must be implementable in the near-term

The SDR codec should serve as the core coding engine for HDR and 360 video

Modifications/customizations can be allowed if justified, e.g., subjective quality improvement

Summary of J0015

- This joint response follows these design considerations

- For SDR:

- Great attention is paid to reduce decoder complexity, many tools are simplified with little performance loss
 - Further performance improvement achieved through 2 additional tools

	BD rate savings			Enc time *	Dec time
	Y	U	V		
CS1	35.72%	44.75%	44.95%	1710%	263%
CS2	27.18%	43.81%	44.51%	1827%	301%

* Encoding time increase mostly due to MTT implementation

- For HDR:

- Improve HDR coding efficiency using coding tools that reside inside the core codec

- For 360 video:

- Main goal is to solve face seam artifacts for this type of video → small modifications are proposed to some SDR tools
 - Exploration of projection format as a “coding tool” for coding performance improvement

- This presentation is organized by categories

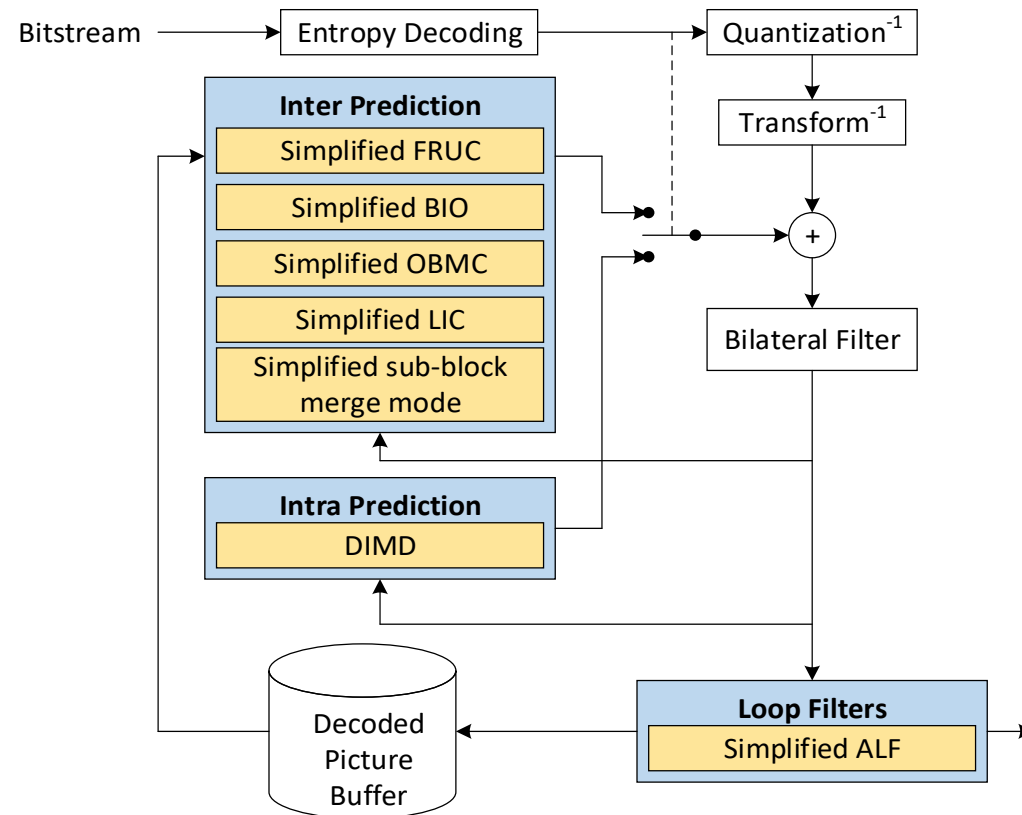
SDR Category

SDR key features

- Decoding complexity reduction
 - Simplified frame-rate up conversion (FRUC)
 - Simplified bi-directional optical flow (BIO)
 - Simplified overlapped block motion compensation (OBMC)
 - Simplified local illumination compensation (LIC)
 - Simplified sub-block merge mode
 - Simplified adaptive loop filter (ALF)
- Coding performance improvement
 - Multi-type tree (MTT)
 - Decoder-side intra mode derivation (DIMD)

SDR Category: simplification of existing JEM tools

SDR key features



Simplified frame-rate up conversion (FRUC)

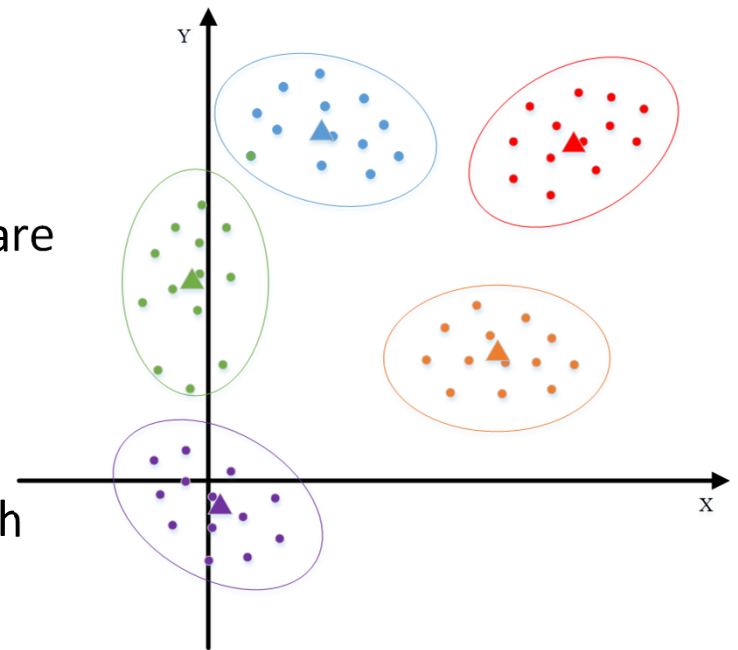
- The FRUC in JEM-7.0 is used as the base
 - Two FRUC modes: template-matching and bilateral-matching
 - Both FRUC modes follow a 4-step process
 - CU-level motion search, CU-level motion refinement, sub-CU-level motion search and sub-CU-level motion refinement
- Proposed FRUC simplifications
 - **MV clustering method** at CU-level and sub-CU-level motion search
 - **Multi-stage motion refinement method** at CU-level and sub-CU-level motion refinement
 - **Early termination** to skip sub-CU-level motion search and sub-CU-level motion refinement
 - **One constrained search range** applied to the entire FRUC process

FRUC: MV clustering based motion search (1/2)

- Initial MV candidates include the following
 - Same as in JEM-7.0
 - AMVP predictors if the CU is coded by AMVP
 - Regular merge candidates
 - Spatial neighboring MVs from the top and left neighbors
 - Different from JEM-7.0
 - The interpolated MV candidates are removed
 - ATMVP and STMVP candidates included only at sub-CU-level motion search
- Reference picture selection
 - Select the reference picture most frequently used by initial MV candidates
 - MV scaling and pruning
 - Scale the MV candidates to the selected reference picture(s)
 - Remove any redundant MV candidates
 - Benefits
 - ***Only one reference picture per L0/L1 list*** need to be accessed during FRUC motion search
 - Reduce the memory bandwidth

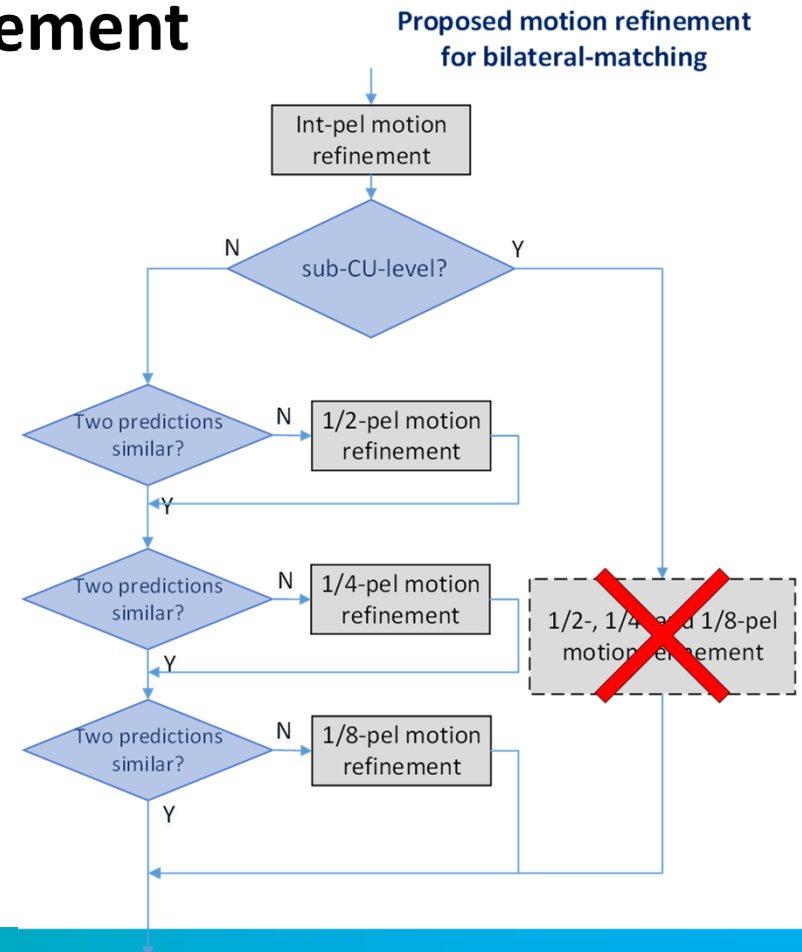
FRUC: MV clustering based motion search (2/2)

- MV clustering
 - Cluster initialization
 - Cluster assignment based on MV similarity
 - Cluster update
 - The centroids of the resulting MV clusters are used for CU-/sub-CU-level motion search
- Advantages
 - Reduce the number of tested MVs
 - The clustering process is performed on each MV candidate only once
 - No iteration is needed, very low complexity



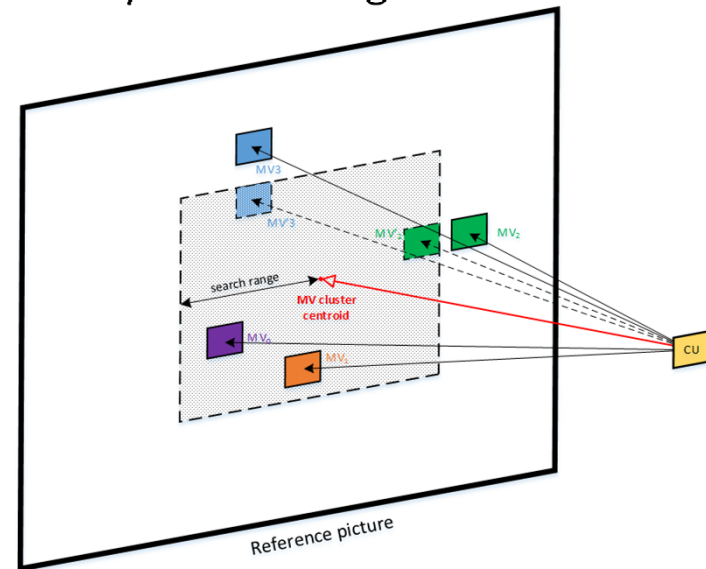
FRUC: Multi-stage motion refinement

- Motion refinement in JEM-7.0
 - MV is searched at 1/4-pel steps
- Proposed multi-stage motion refinement
 - Integer-pel motion refinement
 - Perform integer MV search around the initial MV and update the search center
 - **At most 12 search iterations**
 - Fractional-pel motion refinement
 - Improve the MV precision progressively at 1/2-, 1/4- and 1/8-pel
 - Disable sub-CU-level fractional motion refinement
 - For FRUC bilateral-matching, skip fractional-pel motion refinement based on prediction similarity
 - Template-matching uses a similar multi-stage process
 - Combined with early termination (next slide)



FRUC: Other simplifications

- Early termination of sub-CU-level motion derivation considering the following conditions:
 - If FRUC template-matching
 - If FRUC bilateral-matching and if the two CU-level prediction signals are similar
- Constrained search range for the FRUC
 - One search region of 96×96 luma samples is determined around the centroid of the largest MV cluster
 - Clip each tested MV to be within the search region
 - ***Single memory access to only one reference picture per list is needed***



Simplified Bi-directional optical flow (BIO)

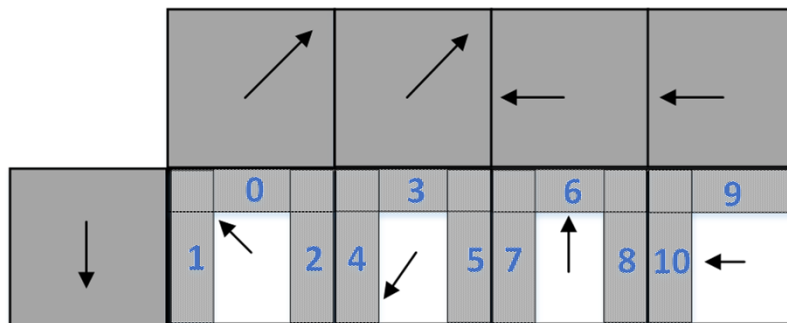
- Two-stage early termination of the BIO
 - Conditionally skip the BIO refinement based on the similarity between two prediction signals at both CU-level and sub-CU-level
- The modified filtering order for the BIO gradient derivation
 - For gradient derivations, horizontal filter is always firstly applied (sample or gradient), followed by vertical filter (gradient or sample)
 - Better aligned with the other filtering processes, such as MC and de-blocking
- The modified rounding operation
 - The BIO in JEM-7.0 uses logical right-shifts
 - Calculate the absolute of input, right-shift the absolute, and multiply the sign
 - Proposed
 - Use arithmetic right-shifts, i.e., the sign is kept during the right-shifts
 - Avoid the need to calculate absolute values

Simplified overlapped block motion compensation (OBMC)

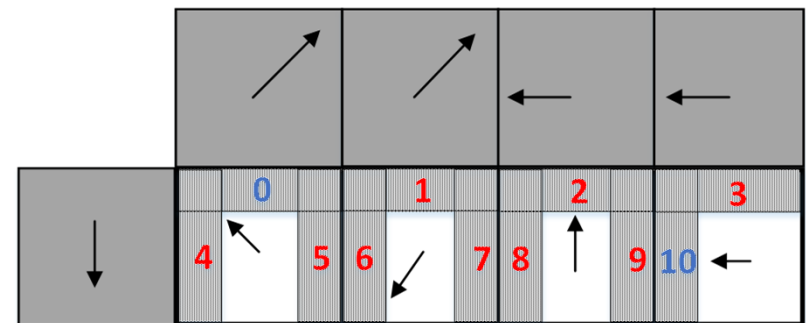
- The OBMC in the JEM-7.0
 - Processing is based on 4x4 block units following raster scan order
 - Only being skipped when neighboring 4x4 blocks have the same motion
- Observations
 - With the existing OBMC order, the MCs of neighboring 4x4 blocks cannot be merged even when they have the same motion
 - OBMC is less effective when neighboring 4x4 blocks have similar motion
- Simplified OBMC
 - **OBMC block grouping** is used to reduce the number of MCs during the OBMC
 - **Early termination** to skip OBMCs when the neighboring blocks have similar motion, i.e., same reference picture(s) and the MV difference is no larger than 1/2-pel

OBMC: Block grouping (1/2)

- Divide the OBMC into two steps
 - **Boundary OBMC:** adjusting the samples on the CU boundaries using the MVs from spatial neighboring CUs
 - **Internal OBMC:** adjusting the samples on the sub-block boundaries, except the samples on the CU boundaries
 - The order of adjusting the boundary samples of the CU is changed



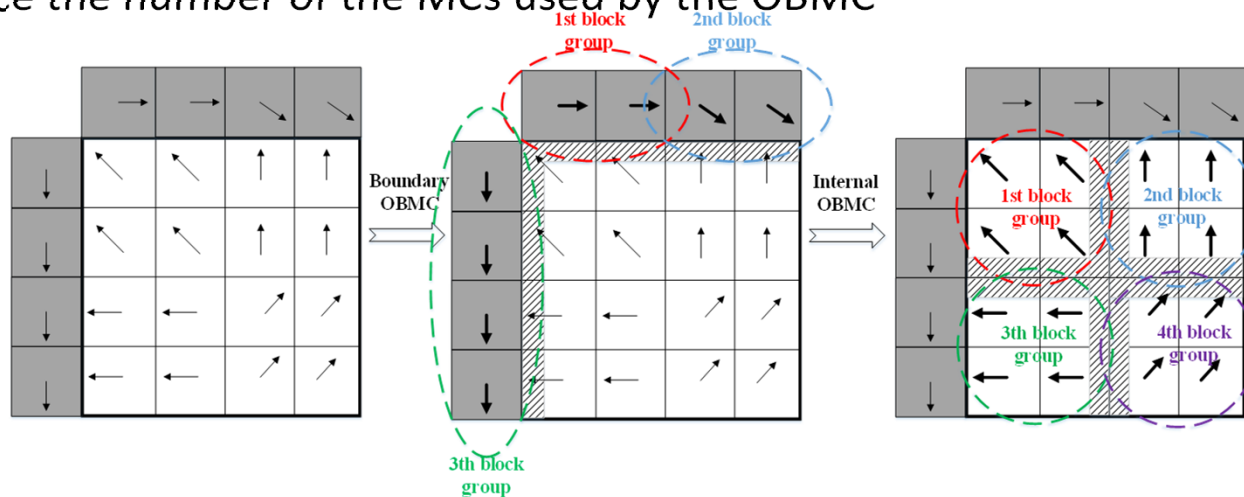
JEM-7.0



Proposed

OBMC: Block grouping (2/2)

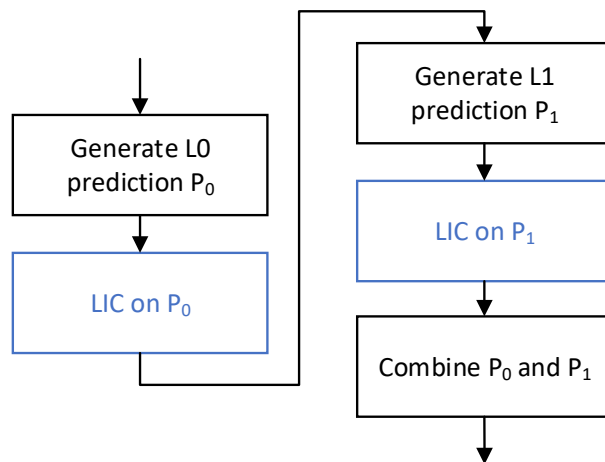
- Block grouping in the OBMC
 - Boundary OBMC: merging the OBMCs of adjacent neighboring 4x4 blocks on the top/left CU boundary with the same motion into one single MC
 - Internal OBMC: aligning the sub-block size used for the OBMC with that used for the sub-block-level motion derivation, e.g., FRUC, ATMVP → larger than 4x4 sub-block size may be used
 - Reduce the number of the MCs used by the OBMC



Simplified local illumination compensation (LIC)

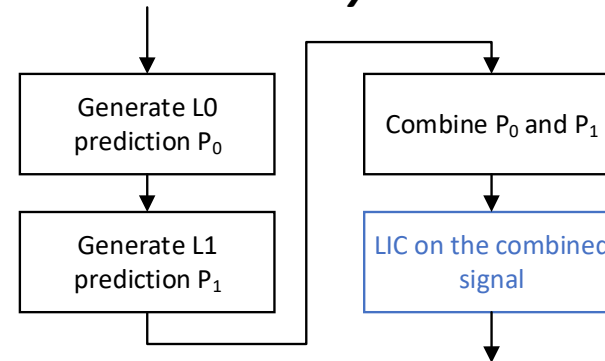
LIC in JEM-7.0

- For bi-prediction, LIC parameters are estimated and applied to L0 and L1 separately
- LIC parameters are estimated at both MC and OBMC



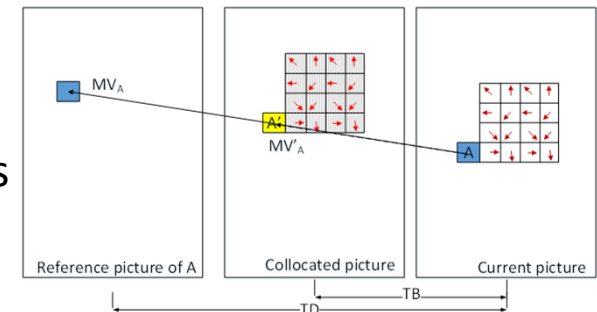
LIC in this response

- LIC parameters are estimated and applied to the bi-prediction signal **only once**
- Reuse the LIC parameters of the MC at OBMC
- **Reduce the worst-case number of LIC derivations by almost 10 times**



Sub-block merge mode simplification

- ATMVP/STMVP in JEM-7.0
 - Uses fixed 4x4 sub-block size
 - Needs to store 4x4 motion field for all reference pictures
- Simplified ATMVP-based motion derivation
 - Use the same collocated picture concept as in HEVC
 - *Only need to access motion from one reference picture (i.e., the collocated picture) for temporal motion derivation*
 - MV candidates are scaled to the collocated picture in the ATMVP motion field
- Add picture/slice-level signaling to support variable sub-block sizes for ATMVP/STMVP-based motion derivation
 - Encoder determines the optimal sub-block size based on the sizes of sub-block merge CUs in the previous picture at the same temporal layer



Adaptive loop filter simplification

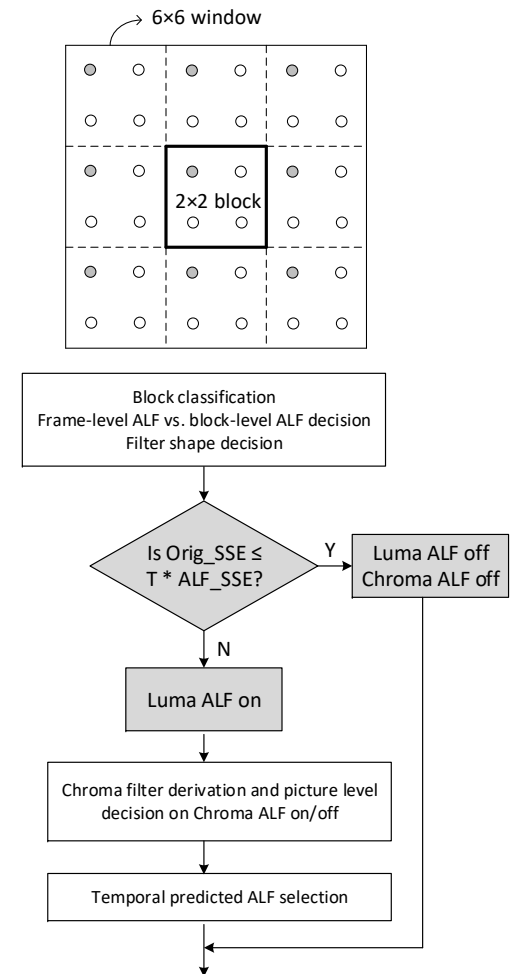
- Simplified block classification
 - For the highest temporal layer pictures, calculate the derivative only at top-left sample position of each 2×2 block
- Encoder-only condition to bias toward disabling ALF for some pictures
 - Luma and chroma ALF are disabled for pictures *in the highest and second highest temporal layers* if:

$$Orig_SSE \leq T \cdot ALF_SSE$$

Orig_SSE: SSE between original picture and reconstructed picture without ALF

ALF_SSE: SSE between the original picture and reconstructed picture with ALF

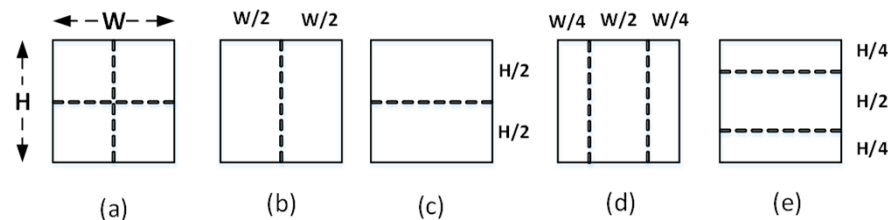
T: bias factor



SDR Category: additional tools not in JEM

Multi-type tree (MTT) partition

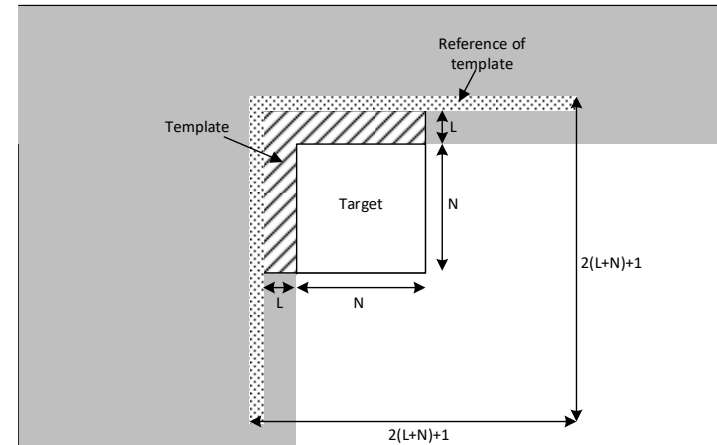
- Besides the quad-tree plus binary-tree (QTBT), triple-tree (TT) from JVET-D0117 is used in this response
 - TT splits one block horizontally/vertically into three parts in a ratio of 1: 2: 1
 - When splitting one QT leaf node, one additional flag is signal to indicate whether it is BT or TT partition
 - Constraints are applied in signaling to avoid redundant partitions



- Fast MTT encoding
 - RD tests of certain repeated partitions are skipped
 - Early termination of horizontal or vertical TT partition based on the RD costs of BT partitions
 - Detailed description available in J0015

Decoder-side intra mode derivation (DIMD)

- DIMD (JVET-C0061 and JVET-D0097) improves intra coding efficiency
 - DIMD flag is signaled at CU level
 - If DIMD, the intra mode is not signaled, and instead derived from neighboring reconstruction samples
 - 129 angular intra directions are implicitly supported
- During parsing, the intra mode of a DIMD CU is unknown, so intra mode propagation (JVET-D0098) is applied to avoid parsing dependency
 - The DIMD intra modes are only derived at reconstruction stage
 - ***Lower parsing complexity than the MPM-based parsing of normal intra mode***



Compression performance

Category	Sequence	HM		JEM	
		CS1	CS2	CS1	CS2
SDR-A	FoodMarket4	35.70%		3.71%	
	CatRobot1	41.50%		3.78%	
	DaylightRoad2	42.87%		4.40%	
	ParkRunning3	34.88%		5.59%	
	Campfire	37.12%		4.44%	
SDR-B	BQTerrace	33.45%	28.90%	4.49%	3.78%
	RitualDance	29.80%	24.32%	3.49%	3.38%
	MarketPlace	30.98%	23.62%	2.65%	3.74%
	BasketballDrive	33.75%	28.26%	3.88%	3.65%
	Cactus	37.09%	30.80%	3.40%	3.65%
Average all		35.72%	27.18%	3.98%	3.64%
Enc Time		1710%	1827%	205%	203%

Decoding complexity: V.S. HM anchor

Category	Sequence	CS1				CS2			
		R1	R2	R3	R4	R1	R2	R3	R4
SDR-A	FoodMarket4	232%	236%	232%	227%				
	CatRobot1	257%	284%	274%	251%				
	DaylightRoad2	231%	241%	251%	257%				
	ParkRunning3	267%	262%	267%	278%				
	Campfire	238%	234%	234%	244%				
SDR-B	BQTerrace	203%	218%	242%	241%	211%	232%	238%	246%
	RitualDance	299%	276%	281%	274%	295%	294%	307%	315%
	MarketPlace	277%	277%	288%	272%	283%	296%	306%	311%
	BasketballDrive	327%	316%	300%	304%	309%	313%	322%	333%
	Cactus	306%	305%	307%	275%	383%	375%	375%	355%
Average		263%				301%			

- Average decoding: 2.63x and 3.01x the HM anchor for CS1 and CS2
- More time saving for CS1 because BIO is more frequently invoked in CS1

Decoding complexity: V.S. JEM anchor

Category	Sequence	CS1				CS2			
		R1	R2	R3	R4	R1	R2	R3	R4
SDR-A	FoodMarket4	27%	28%	29%	32%				
	CatRobot1	27%	28%	28%	29%				
	DaylightRoad2	30%	31%	33%	35%				
	ParkRunning3	33%	32%	33%	35%				
	Campfire	31%	35%	38%	43%				
SDR-B	BQTerrace	26%	28%	32%	33%	39%	45%	46%	49%
	RitualDance	37%	39%	41%	44%	44%	46%	50%	52%
	MarketPlace	31%	33%	35%	36%	41%	41%	44%	46%
	BasketballDrive	36%	38%	41%	44%	44%	47%	49%	54%
	Cactus	30%	31%	32%	32%	39%	41%	45%	48%
Average		33%				45%			

- Average decoding: 0.33x and 0.45x the HM anchor for CS1 and CS2

“Tool-on” and “Tool-off” tests

- Decoding complexity reduction
 - **FRUC SIMP**: simplifications on FRUC
 - **BIO SIMP**: simplifications on BIO
 - **OBMC SIMP**: simplifications on OBMC
- Coding performance improvement
 - **MTT**: multiple-type tree
 - **DIMD**: decoder-side intra mode derivation

Tested tool	“Tool-off”			“Tool-on”		
	BD-rate	Run time		BD-rate	Run time	
		Enc	Dec		Enc	Dec
FRUC SIMP	0.00%	105%	156%	0.14%	95%	67%
BIO SIMP	-0.17%	104%	146%	0.13%	95%	79%
OBMC SIMP	-0.03%	104%	120%	0.05%	93%	87%
MTT	4.35%	43%	101%	-3.26%	236%	102%
DIMD	0.41%	90%	99%	-0.41%	107%	100%

These simplifications reduce decoding complexity as well as encoding complexity

Main reason for encoding time increase

Further simplifications after submission

- Further simplification on LIC
 - In the CfP submission, LIC parameter derivation is simplified, but the LIC sample adjustment is still applied at both MC and OBMC stages
 - After CfP submission, LIC is further simplified to apply the LIC sample adjustment only once on the prediction signal after OBMC
 - Significantly reduce both average and worst-case LIC complexity without any performance loss
- Latency reduction on template-based inter prediction (JVET-J0045)
 - Two template-based inter prediction modes in JEM-7.0
 - FRUC template-matching and LIC
 - Significant decoding architecture issue because the reconstructed samples of spatial neighbors are needed in these template-based inter modes
 - Inhibits the ability to parallel decode CTUs or CUs
 - Proposed solution in J0045:
 - Use prediction samples and add reconstructed DC component only to form the template samples
 - Keep majority of the coding gain of these two modes: only lose 0.46% of the total 3.27% BD rate savings from these two tools

SDR summary

- Design goals of this CfP response
 - Reduce average and worst-case decoding complexity without causing performance loss
 - Further improve the coding efficiency beyond JEM
- The response provides good candidate technologies to achieve the goals
 - Average coding efficiency about 35.7% and 4% better than HM and JEM anchors
 - Average decoding time about 2.6x and 0.3x the decoding time of HM and JEM
 - Tool-on and tool-off tests show significant complexity reduction and negligible performance loss from the proposed simplifications
- The proposed technologies are suitable to be included in the Test Model

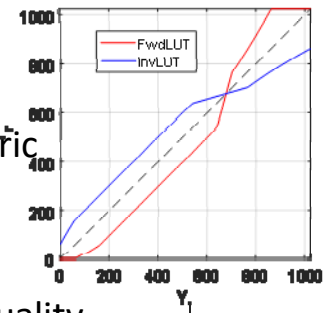
HDR Category

HDR Summary

- Two major HDR specific coding tools
 - In-loop reshaping
 - Luma-based QP prediction
- Pre- and Post-processing are unnecessary
 - All proposed tools reside inside core codec
 - Keep conformance point same as SDR
- HDR CfP submitted bitstreams target better subjective performance
 - In-loop resaper can also be configured to maximize HDR objective metrics

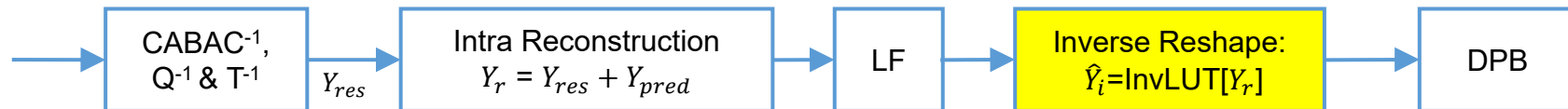
In-loop Reshaper: Luma Only

- Reshaper: A pair of one dimensional, 10-bit, 1024-entry mapping tables (1D-LUT): FwdLUT and InvLUT
 - Remaps Luma code values inside the coding loop for better HDR coding efficiency
 - Reconstructed pictures in DPB is in the same form as original source picture
- Design philosophy :
 - Default mode:
 - Uses a pre-defined Luma mapping curve derived from weights from the wPSNR metric
 - Helps validate the in-loop reshaping architecture
 - Adaptive mode:
 - Allows adaptive adjustment to the Luma mapping curve for better HDR subjective quality
 - Enables future commercial optimization using better HVS models
 - Pictures stored in the DPB are in the input signal domain



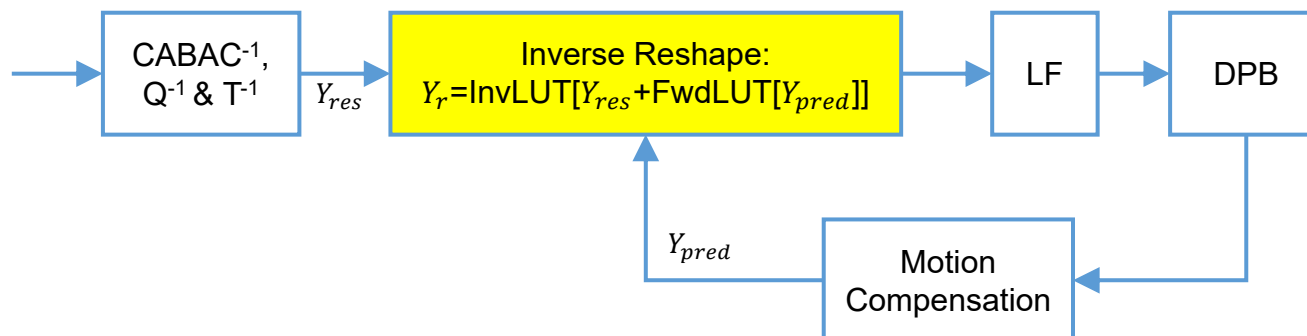
In-loop Reshaper: Intra Slice

- Coding process is performed in the reshaped domain, followed by inverse reshaping before storing in DPB
- Encoder:
 - Forward reshaping, intra coding in reshaped picture, inverse reshaping reconstructed picture before stored in DPB buffer.
- Decoder:
 - Intra decoding, inverse reshaping reconstructed picture before stored in DPB buffer



In-loop Reshaper: Inter Slice

- Reshaping is performed within the prediction loop
- Encoder:
 - Modify how residue is generated: $Y_{res} = FwdLUT[Y_i] - FwdLUT[Y_{pred}]$
- Decoder:
 - Modify how reconstructed sample is generated:
 $Y_r = InvLUT[Y_{res} + FwdLUT[Y_{pred}]]$



Luma Reshaper LUT

- Default reshaper: pre-defined luma mapping tables based on wPSNR
- Adaptive reshaper: use parametric model for codeword distribution
 - Divides Luma channel into 32 bins or intervals.
 - Parametric model:
 - signals picture profiling and bin importance
 - pre-defines how codewords are allocated for Luma intervals based on profile and bin importance.
 - Typical size of reshaper syntax is around 30 bits.
- In CfP submission, adaptive reshaper is applied and syntax is only sent in intra slices.

In-loop Reshaper: Additional Configurable Coding Tool

- Region-of-Interest reshaping:
 - Reshaping is only enabled in ROI
 - ROI is defined as a rectangular area in multiples of CTU size
 - For intra slice, to avoid boundary artifacts, inverse reshaping is applied to ROI before loop filter
 - Used on HLG content in the submission
- Luma-based Chroma QP Offset
 - To compensate for Luma signal interaction with Chroma signal, an additional Chroma QP Offset is used to adjust Chroma QPs
- Loop Filter Optimization
 - For inter slice, control if loop filter is applied in reshaped domain or not

Luma-based QP Prediction

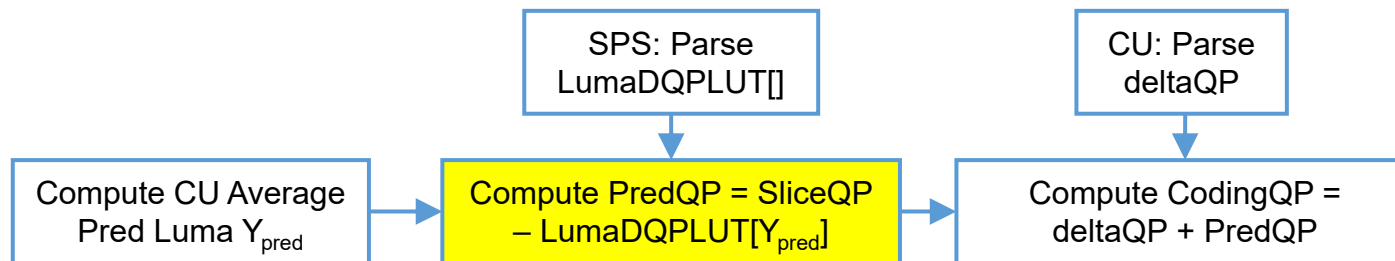
- Target: reduce bitrate overhead of the deltaQP syntax when spatial Luma-dependent QP adaptation (LumaDQP) is used.
- Assumption: CU average predicted luma value is very close to that of original
- Encoder:

$$\text{CodingQP} = \text{SliceQP} - \text{LumaDQPLUT}[\text{Yorg}]$$

$$\text{PredQP} = \text{SliceQP} - \text{LumaDQPLUT}[\text{Ypred}]$$

$$\text{deltaQP} = \text{CodingQP} - \text{PredQP} = \text{LumaDQPLUT}[\text{Ypred}] - \text{LumaDQPLUT}[\text{Yorg}]$$

- Decoder:



Additional Encoder Operation

- Joint optimization of reshaper and luma QP adaptation
- RDO in Mode Decision:
 - intra: SSE
 - inter: wSSE
- Chroma QP offset adjustment
 - PQ: same as anchor
 - HLG: chromaQPoffset is derived in a similar way as that used for the PQ anchors

Compression Performance:

- Setting:
 - RA configuration similar to SDR setting
 - Both HDR specific coding tools and all SDR coding tools are applied.
- CfP submission: adaptive reshaping
- Additional results: default reshaping

CfP Submission: Compared to HM and JEM

		Over HM anchor					Over JEM-7.0 anchor				
		BD-rate					BD-rate				
		DE100	PSNRL100	wPsnrY	wPsnrU	wPsnrV	DE100	PSNRL100	wPsnrY	wPsnrU	wPsnrV
HDR-A	DayStreet	-59.0%	-34.6%	-26.8%	-58.2%	-71.5%	-2.0%	2.9%	13.5%	-7.1%	-8.6%
	PeopleInShoppingCenter	-51.7%	-27.6%	-23.2%	-62.2%	-58.3%	-3.7%	2.5%	7.1%	-12.9%	-11.0%
	SunsetBeach	-60.4%	-28.1%	-22.7%	-88.3%	-88.0%	-10.0%	-2.7%	-2.0%	-14.9%	-22.8%
HDR-B	Market	-29.7%	-28.4%	-24.3%	-31.1%	-24.2%	15.7%	-5.3%	1.0%	27.2%	36.8%
	ShowGirl2	-25.1%	-23.7%	-16.8%	-50.1%	-22.8%	4.0%	3.8%	15.0%	27.5%	15.4%
	Hurdles	-42.3%	-39.8%	-33.1%	-54.9%	-47.4%	13.9%	-8.1%	0.7%	17.4%	18.9%
	Starting	-38.7%	-30.3%	-24.6%	-26.7%	-30.7%	-7.7%	0.8%	8.6%	12.4%	-0.7%
	Cosmos1	-32.8%	-30.0%	-17.7%	-61.3%	-76.6%	-3.5%	-7.8%	8.8%	14.8%	1.2%
Average HDR-A		-57.0%	-30.1%	-24.2%	-69.6%	-72.6%	-5.2%	0.9%	6.2%	-11.7%	-14.1%
Average HDR-B		-33.7%	-30.4%	-23.3%	-44.8%	-40.4%	4.5%	-3.3%	6.8%	19.8%	14.3%
Average all		-42.5%	-30.3%	-23.7%	-54.1%	-52.4%	0.8%	-1.7%	6.6%	8.0%	3.6%
Enc Time[%]		1462%					183%				
Dec Time[%]		292%					33%				

Additional Results: Default reshaper

Anchor HM: BD-rate								
	DE100	PSNRL100	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV
Market	-38.8%	-28.8%	-28.8%	-41.5%	-43.2%	-25.4%	-44.7%	-43.1%
ShowGirl2	-38.0%	-29.7%	-30.6%	-69.3%	-52.7%	-27.7%	-70.6%	-56.6%
Hurdles	-47.5%	-38.9%	-37.7%	-57.8%	-53.1%	-34.0%	-54.0%	-49.4%
Starting	-36.7%	-34.4%	-33.0%	-36.0%	-34.9%	-30.3%	-37.3%	-30.0%
Cosmos1	-36.4%	-26.4%	-26.2%	-71.1%	-83.8%	-24.2%	-70.9%	-83.9%
Average HDR-B	-39.5%	-31.6%	-31.3%	-55.1%	-53.6%	-28.3%	-55.5%	-52.6%

Anchor JEM: BD-rate								
	DE100	PSNRL100	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV
Market	1.7%	-5.9%	-5.2%	8.9%	4.2%	-2.6%	-8.9%	-16.5%
ShowGirl2	-15.2%	-4.5%	-4.4%	-23.4%	-35.2%	-2.7%	-34.0%	-46.0%
Hurdles	4.6%	-7.0%	-6.5%	6.7%	11.3%	-4.7%	3.2%	8.1%
Starting	-3.8%	-5.3%	-4.0%	-2.5%	4.4%	-2.4%	-8.0%	-9.0%
Cosmos1	-8.7%	-3.0%	-2.8%	-13.9%	-30.7%	-1.9%	-14.4%	-29.3%
Average HDR-B	-4.3%	-5.1%	-4.6%	-4.8%	-9.2%	-2.9%	-12.4%	-18.5%

Subjective Results: Market @R2



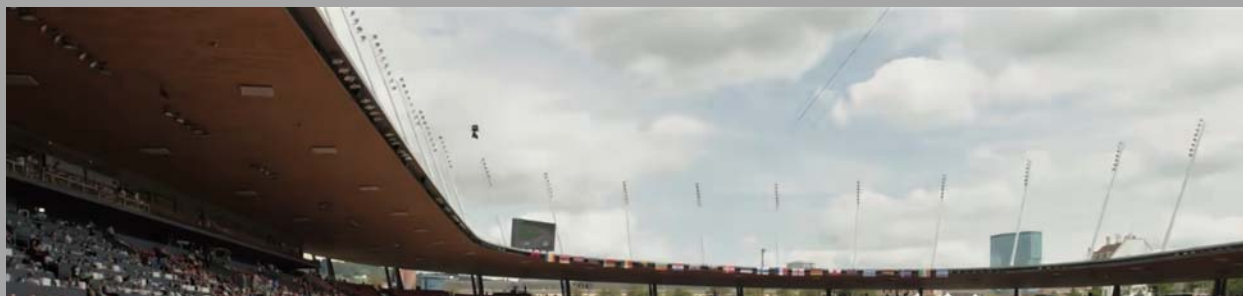
JEM anchor



Response

© 2018 InterDigital, Inc. and Dolby Laboratories, Inc. All Rights Reserved.

Subjective Results: Starting@R1



JEM anchor



Response

In-loop Reshaper Tool-Off Test

- Default reshaper is configured to perform the test.
- The anchor of the “tool-off” test is the CfP response using default in-loop reshaper.

		BD-rate							
		DE100	PSNRL100	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV
HDR-B	Market	7.1%	7.8%	5.1%	15.1%	12.3%	-1.2%	5.2%	4.4%
	ShowGirl2	5.6%	2.9%	1.6%	9.2%	12.7%	-1.9%	-1.0%	1.1%
	Hurdles	8.5%	6.5%	5.0%	11.8%	5.8%	2.6%	7.0%	4.5%
	Starting	18.2%	6.0%	2.3%	10.6%	14.1%	-1.0%	4.8%	5.8%
	Cosmos1	1.6%	3.4%	1.3%	-4.5%	-4.4%	-0.1%	-6.5%	-9.2%
	Average all	8.2%	5.3%	3.1%	8.4%	8.1%	-0.3%	1.9%	1.3%
Enc Time[%]		96%							
Dec Time[%]		94%							

HDR summary

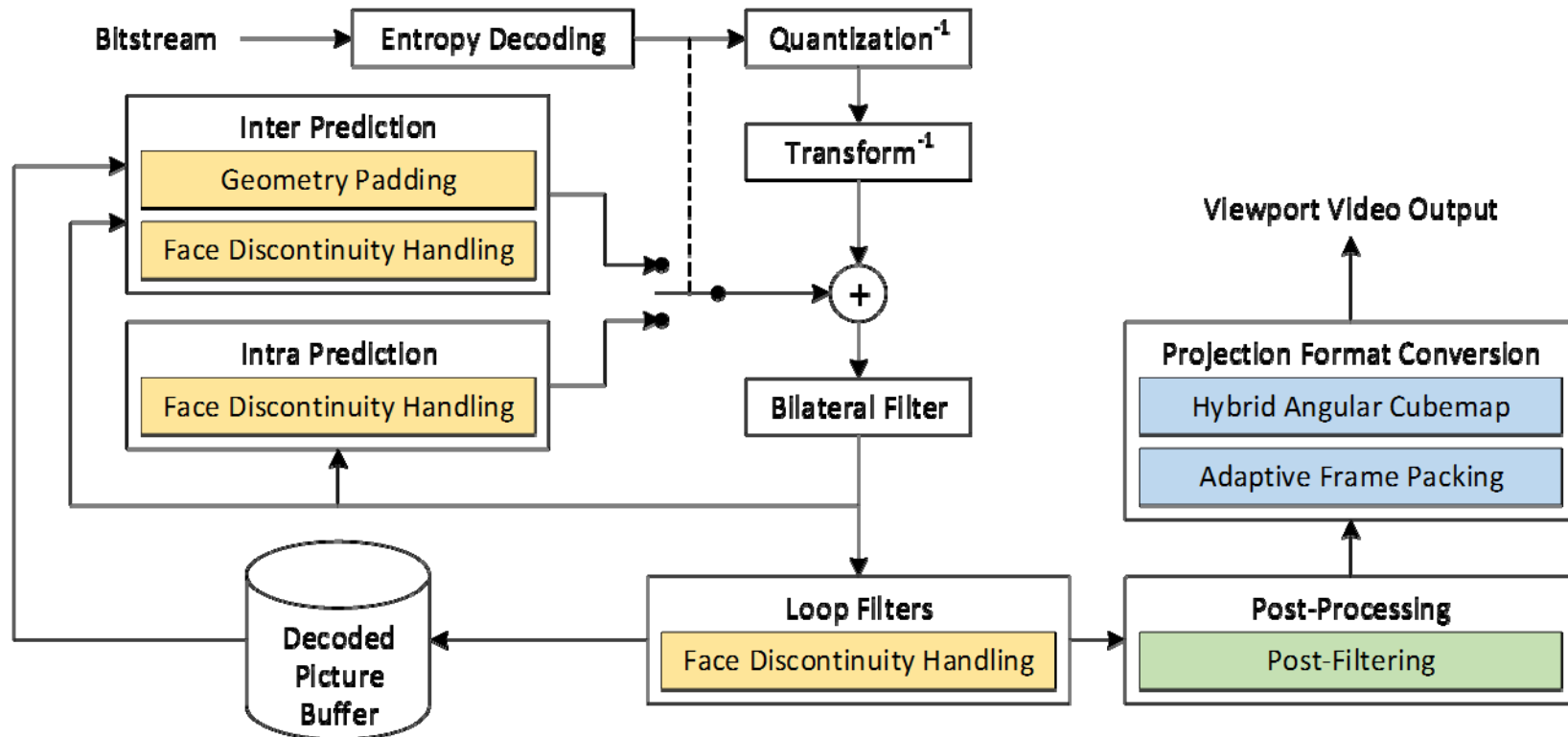
- HDR tools should be part of the base codec:
 - Wide commercial success of HDR in the market place.
- Concept of the reshaper has proven to be beneficial
 - Increasing compression efficiency.
- Tools:
 - Simplified method includes the in-loop reshaper in the core codec
 - Enhanced luma-based QP prediction mechanism.
- Propose inclusion of tools in Test Model.

360° Category

Key Features

- Projection format
 - Hybrid Angular Cubemap (HAC)
 - Adaptive Frame Packing (AFP)
- Coding tools
 - Geometry Padding of Reference Pictures (GP)
 - Face Discontinuity Handling (FDH)
- Post-processing
 - Post-Filtering

360° Video Decoder Block Diagram

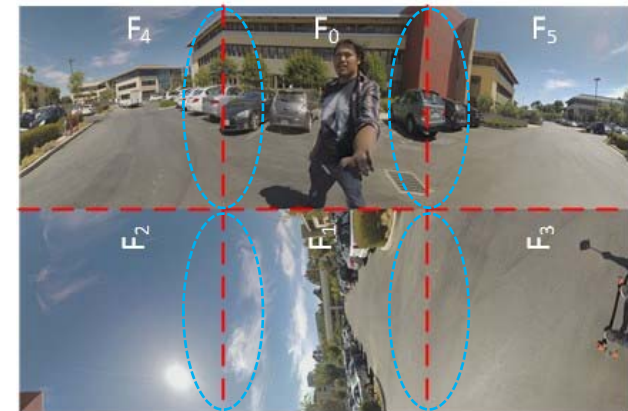


Hybrid Angular Cubemap

- HAC generalizes the EAC projection format using adaptive mapping functions
 - p is the HAC parameter; $p \in [0.3, 1.5]$
 - $p = 1 \Rightarrow \text{HAC} = \text{EAC}$
 - $p \rightarrow 0 \Rightarrow \text{HAC} \rightarrow \text{CMP}$
- Parameter derivation process is done row-by-row
 - 2 parameters for each face: horizontal and vertical directions
 - Constraint is applied to keep the vertical continuity of neighboring faces in the same face row $\rightarrow$ 1 vertical parameter per face row
 - Iterative search for 8 parameters, target is to minimize the conversion only E2E WS-PSNR
- HAC parameters are derived at each IRAP and signaled in the PPS

Forward mapping function from CMP to HAC in one direction

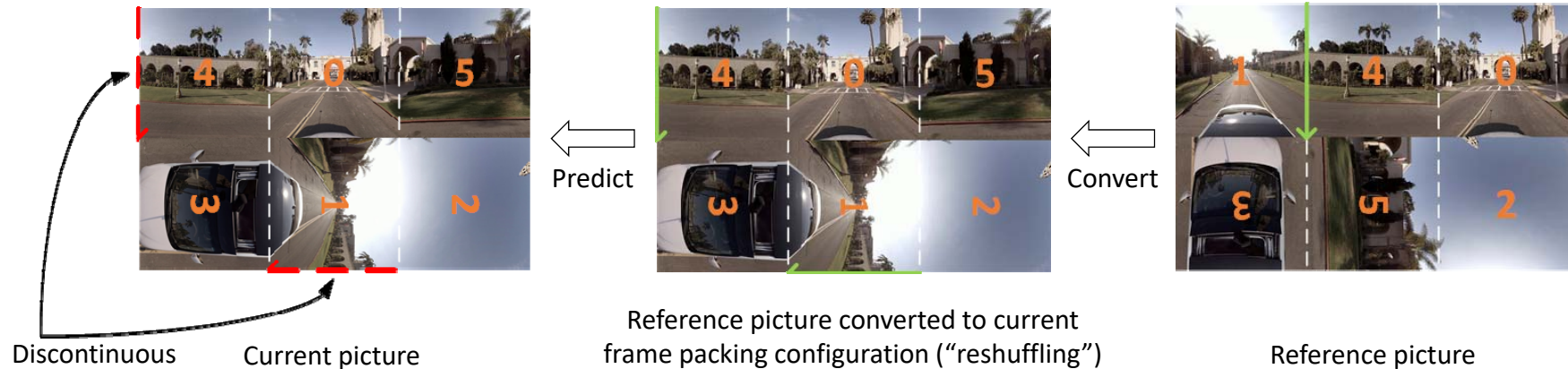
$$\beta' = f(\beta) = \frac{4}{\pi \cdot p} \tan^{-1} \left(\beta \cdot \tan \left(\frac{\pi}{4} \cdot p \right) \right)$$



$$p_4^v = p_0^v = p_5^v \quad p_2^v = p_1^v = p_3^v$$

Adaptive Frame Packing

- Frame packing generates discontinuities, which often lead to visible seam artifacts in the reconstructed viewport
- AFP alternates between (up to) 4 different frame packing configurations at each IRAP
 - Reshuffles the positions of the faces in the frame packed picture
 - Changes the position of discontinuous face boundaries
 - Improves prediction near discontinuous face boundaries → reduced seam artifacts
- AFP parameters are derived at each IRAP and signaled in the PPS
 - Detailed AFP parameter determination method at the encoder is described in J0015

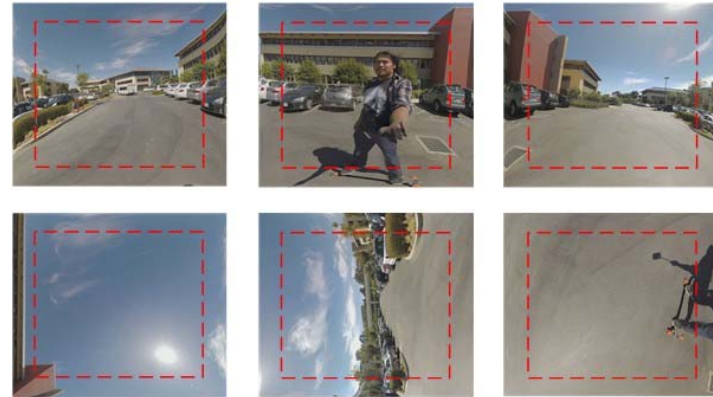


Geometry Padding of Reference Pictures

- Repetitive padding in MCP is not suitable for 360° video, since 360° video is intrinsically all-around and has “no boundaries”
- Geometry padding extends the reference picture boundaries by considering the 3D geometry
 - The proposed method pads each face with meaningful neighboring samples in the spherical domain
 - 144 sample lines are padded outside of each side of each face
- Geometry padding of reference pictures can be implemented on-the-fly in practical decoders
 - Perform padding only if reference samples needed for MCP are outside of the face boundaries
 - Reduces decoding complexity (memory and calculation)



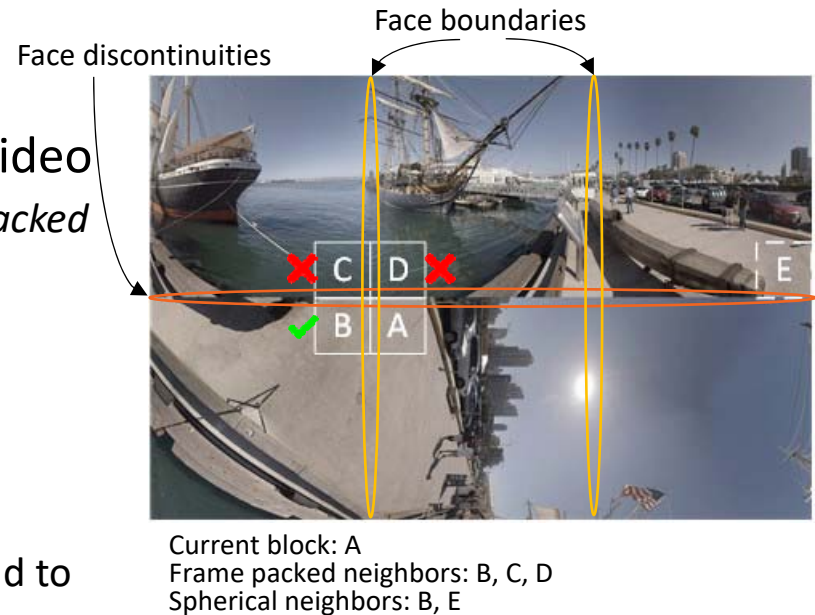
Extended picture of HAC 3×2 with repetitive padding



Extended face pictures of HAC 3×2 with geometry padding

Face Discontinuity Handling

- Two kinds of neighboring blocks in 360° video
 - Neighbors in 2D frame packed picture: *frame packed neighbor*
 - Neighbors in 3D geometry: *spherical neighbor*
- A neighbor in the frame packed picture may not necessarily be the actual neighbor in 3D geometry
 - Using “wrong” neighbors during coding may lead to visible seam artifacts in rendered viewports
- Some tools in the SDR core codec are modified to use information from frame packed neighbors *only if they are also spherical neighbors*
 - Intra prediction, MPM, decoder-side intra mode derivation, MVP, merge mode, FRUC, deblocking, SAO, ALF, OMBC, LIC



Post-Filtering

- PERP uses padding and blending to reduce face seam artifacts
 - This method increases the number of coded samples
 - Seam artifact may still be visible (e.g., in Chairlift)
- Our solution: filter the discontinuous face boundaries using the reconstructed samples in a post processing step
 - The number of coded samples does not increase
 - HEVC deblocking filter with fixed boundary strength = 2 is used



Performance

- In our CfP response submission, we set HAC face size to 1184×1184, the same as used in the 360° video CTC for CMP and EAC
- Some responses (J0014/J0021/J0023) used larger face sizes of 1280
 - Using the same larger face size, our gain can be increased to -34.43% and -14.45% for HM and JEM, respectively

	Over HM anchor			Over JEM anchor		
	BD-rate(Y)	BD-rate(U)	BD-rate(V)	BD-rate(Y)	BD-rate(U)	BD-rate(V)
Balboa	-43.85%	-53.01%	-57.54%	-13.49%	-18.51%	-19.80%
Chairlift	-47.19%	-62.52%	-60.10%	-23.98%	-25.78%	-28.57%
KiteFlite	-25.41%	-55.89%	-61.71%	-9.61%	-15.82%	-20.15%
Harbor	-28.92%	-56.21%	-56.47%	-10.82%	-20.05%	-19.52%
Trolley	-23.97%	-42.56%	-48.14%	-9.65%	-11.30%	-16.76%
Average 360	-33.87%	-54.04%	-56.79%	-13.51%	-18.29%	-20.96%
Enc Time	1332%			241%		
Dec Time	350%			57%		

E2E WS-PSNR results are shown in the table, E2E SPSNR-NN results are similar

Face
Discontinuity
Handling
disabled

Face
Discontinuity
Handling
enabled



Original



HM anchor @R1



JEM anchor @R1



Response @R1





Original



HM anchor @R2



JEM anchor @R2



Response @R2

Original



HM anchor @R3



JEM anchor @R3



Response @R3



360 summary

- An efficient projection format should be used for 360° video coding
 - Allow formats used for capture, storage, coding, and display to be different
- Geometry padding of reference pictures improves visual quality and compression efficiency
 - Higher gains for moving content, which is also a more realistic use case
 - Can be combined with PERP to further improve visual quality (JVET-J0044)
- SDR tools that could cause seam artifacts should be modified for 360° video
 - Padding, guard bands, blending, etc. are only “band-aid” solutions
 - Such modifications can be relatively simple, as shown by the proposed FDH methods
 - Visual quality improvements should be the main target, because seam artifacts are not captured by objective metrics
- Propose to include 360° video coding tools in the Test Model

Closing remarks

- This response achieves large coding efficiency gain and substantial subjective quality improvement in all three categories while maintaining reasonable decoder complexity
- For SDR
 - Great attention is paid to ensure reasonable decoding complexity without losing performance
 - Average-case and worst-case computation complexity, memory bandwidth, etc.
 - For CS1, decoding complexity is $\sim 2.6x$ vs. HM anchor ($\sim 0.3x$ vs. JEM), with 35.7% performance improvement over HM
- For HDR and 360° video
 - Due to wide commercial success of HDR in the marketplace, HDR-specific tools should be part of the base codec
 - For 360° video, certain coding tools should be modified to ensure good subjective quality and avoid face seam artifacts

THANK YOU