

JVET-J0022 - SDR, HDR and 360° video coding technology proposal

Qualcomm and Technicolor / Medium complexity version



MTT codec is proposed in JVET-J0021 and JVET-J0022

- Addressing the SDR, HDR and VR categories
- 3 CfP responses, corresponding to 3 complexity profiles

JVET contrib	Profile	SDR	HDR	VR
J0021	Low Complexity	QTBT	DRA	ACP
J0022	Medium Complexity	QTBT+ABT	DRA + Refinement	PERP + AQP
J0021	High Complexity	QTBT+TT+ABT	DRA	ACP

- MTT contains most of the JEM tools, plus further adaptations / additional features
- Common software platform (MTT)
 - Also submitted in a light version in JVET-J0075 - HM-like + QTBT + TT + ABT

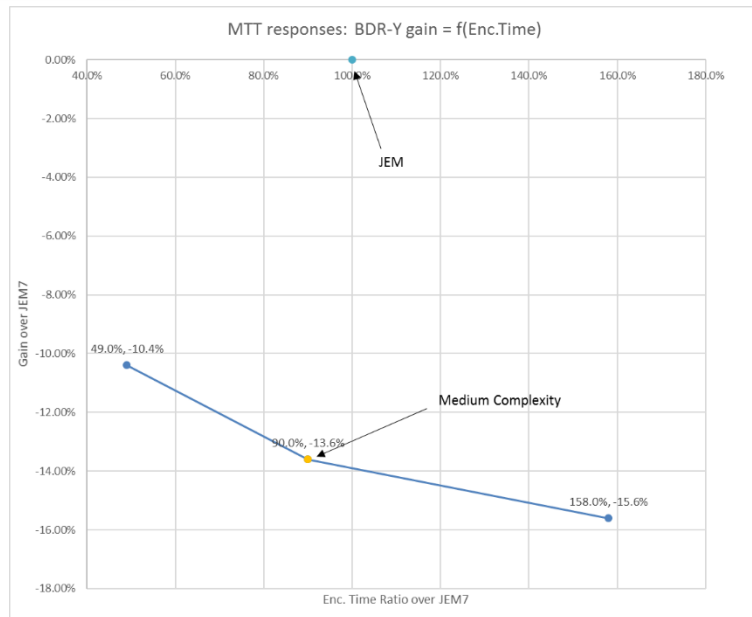
Performance summary of the Medium Complexity response

SDR	Over HM anchors					Over JEM anchors				
	BD-rate PSNR			Time variation		BD-rate PSNR			Time variation	
	Y	U	V	Enc	Dec	Y	U	V	Enc	Dec
CS1	-41.9%	-44.8%	-46.1%	728%	582%	-13.6%	-3.8%	-5.6%	90%	82%
CS2	-33.8%	-51.6%	-52.5%	1035%	524%	-12.7%	-16.7%	-18.8%	119%	89%

HDR	Over HM anchors				Over JEM anchors			
	BD-rate HDR metrics		Time variation		BD-rate HDR metrics		Time variation	
	PSNRL	DE100	Enc	Dec	PSNRL	DE100	Enc	Dec
CS1	-38.3%	-52.2%	528%	604%	-13.3%	-17.0%	96%	90%

360	Over HM anchors					Over JEM anchors				
	BD-rate WS-PSNR			Time variation		BD-rate WS-PSNR			Time variation	
	Y	U	V	Enc	Dec	Y	U	V	Enc	Dec
CS1	-32.1%	-51.6%	-53.6%	579%	519%	-11.7%	-13.1%	-14.1%	97%	102%

Random Access (CS1)



Main tools or adaptations proposed in JVET-J0022

■ Picture topology:

- Asymmetric Binary Tree (ABT)
- Block partitioning on picture border

■ Intra tools:

- Multi-reference intra prediction
- Bi-directional intra prediction

■ Inter tools:

- Unicity in motion information candidate lists
- Extended template merge modes
- Extended affine motion compensation
- Template affine mode
- Generalized OBMC
- Bi-directional illumination compensation

■ Transforms:

- Reduced pool of primary transforms in EMT
- Interaction constraints on EMT/NSST transforms

■ Loop filtering:

- SAO palette

■ Encoder:

- Fast coding methods
- Deep-learning based prediction of split modes in Intra slices, coupled with heuristics
- Various ABT/coding tree configurations to target low, medium and high efficiency

■ Performances:

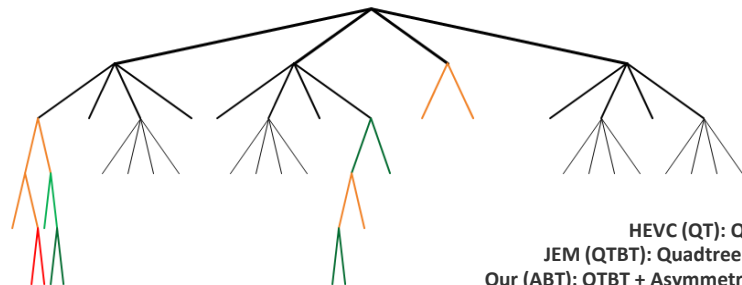
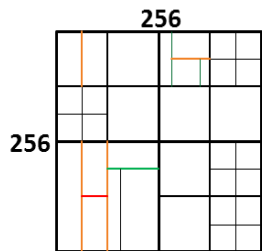
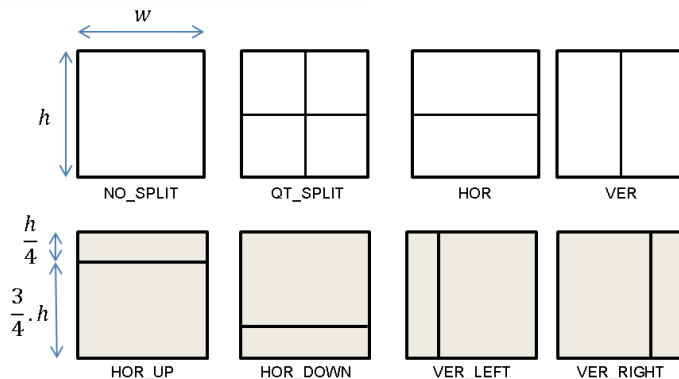
- Range of trade-offs enc/dec time and BD rate gain

Tools description - SDR category

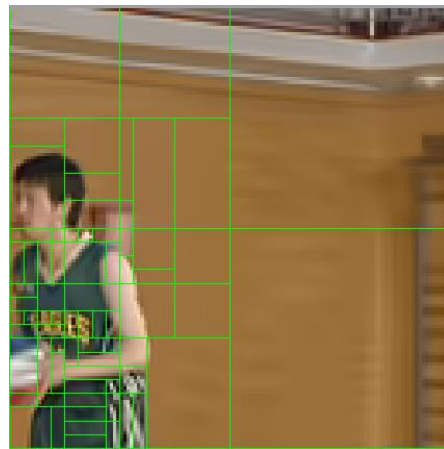


■ QTBT+ABT coding structure (JVET-D0064)

- The Binary Tree can be symmetric or asymmetric: QTBT+ABT (Asymmetric Binary Tree)



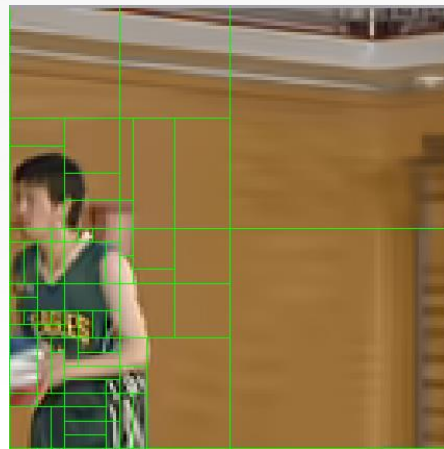
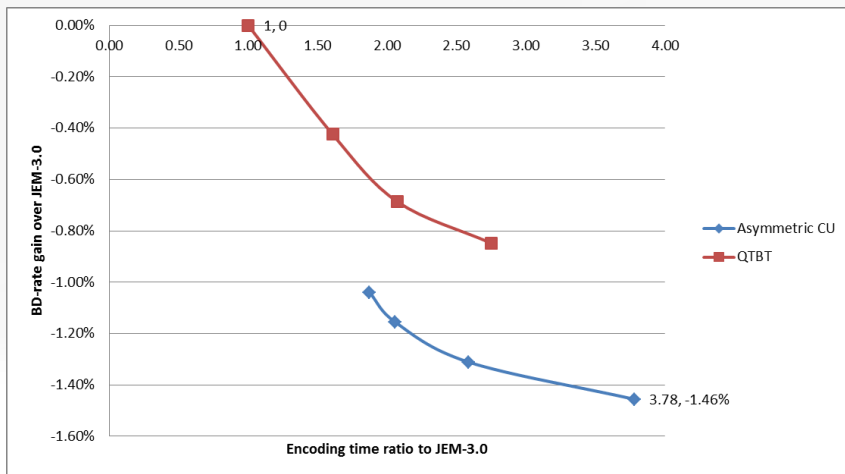
HEVC (QT): Quadtree Tree
JEM (QTBT): Quadtree + Binary tree
Our (ABT): QTBT + Asymmetric Binary tree



New luma block sizes: 12, 24, 48

■ QTBT+ABT coding structure

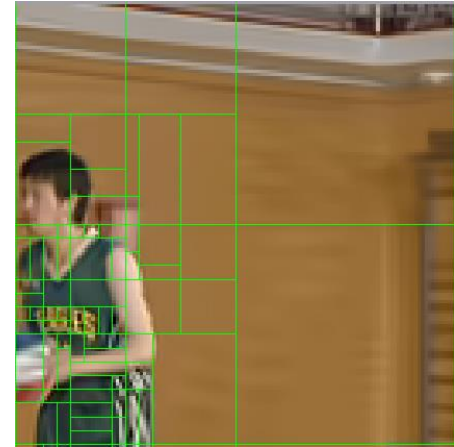
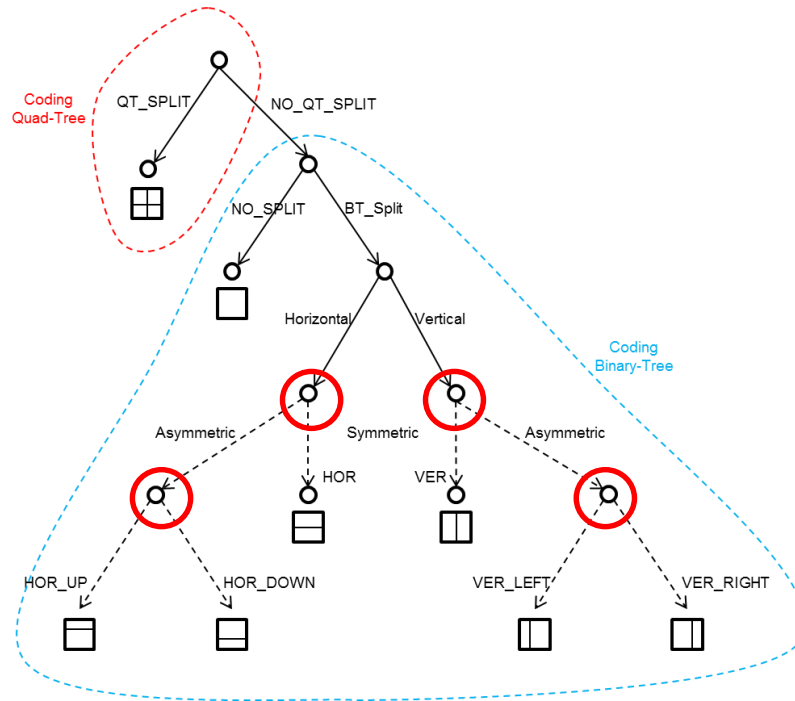
- The Binary Tree can be symmetric or asymmetric: QTBT+ABT (Asymmetric Binary Tree)
- Tool off/tool on:
 - Gain: -3.8%/-3.2% (Y)
 - Enc.time 186%/190%, Dec. time 99%/101%
- Encoding time/BD-rate behavior compared to QTBT



Picture Topology

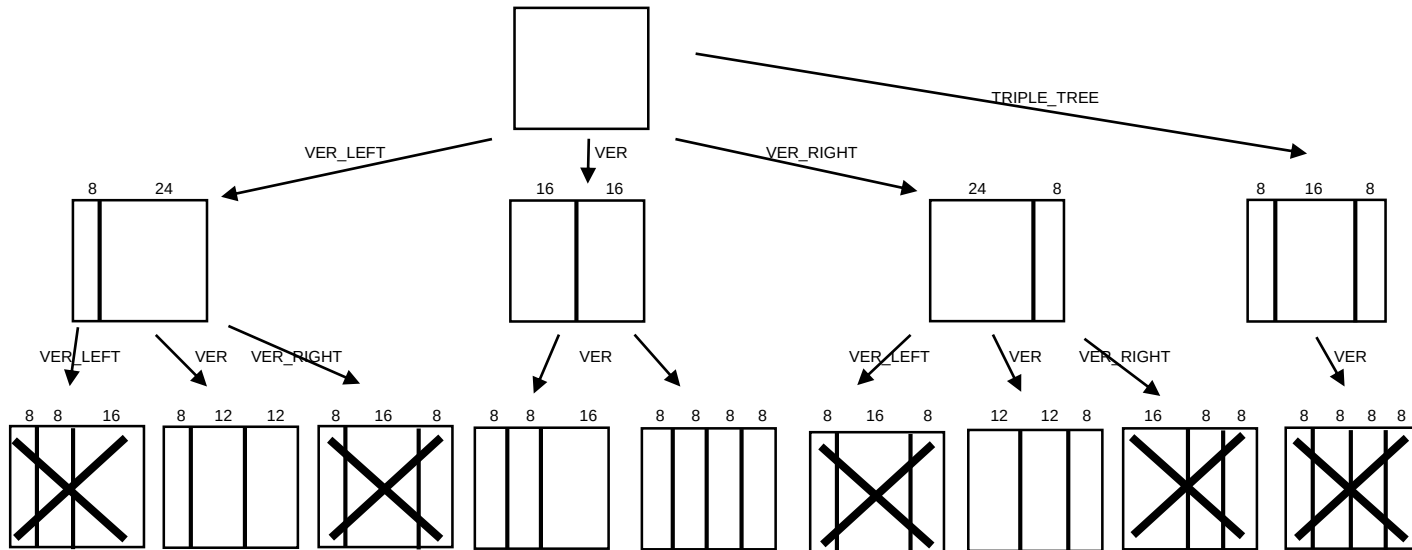
■ QTBT+ABT coding structure

- Syntax to signal the coding tree: 2 additional bins



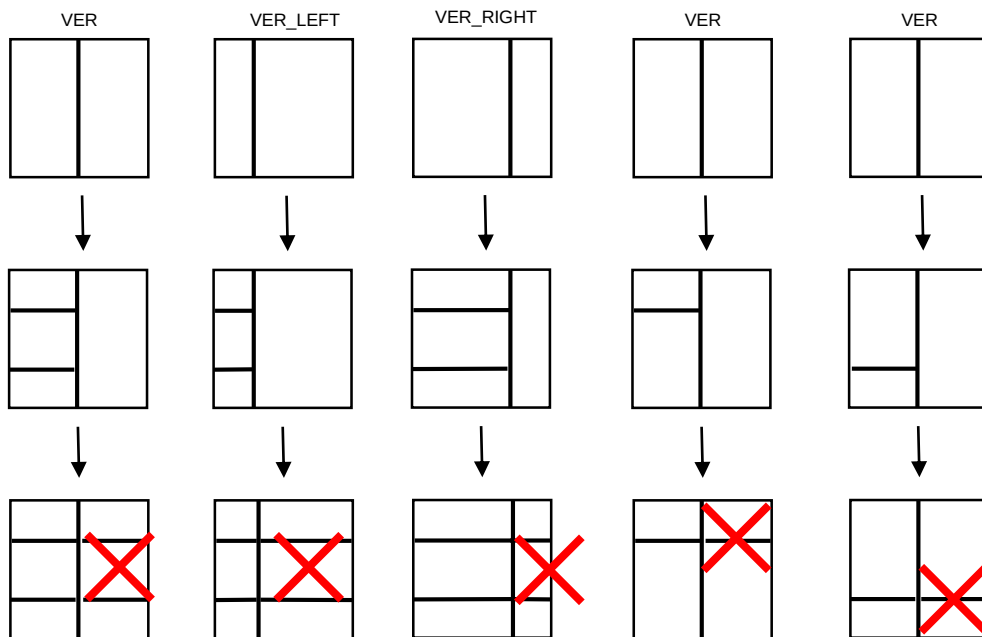
Non-redundancy policy between various successions of splits

- Anti-emulation policy to make each topology reachable through at most one sequence of quad-tree/binary/triple splits



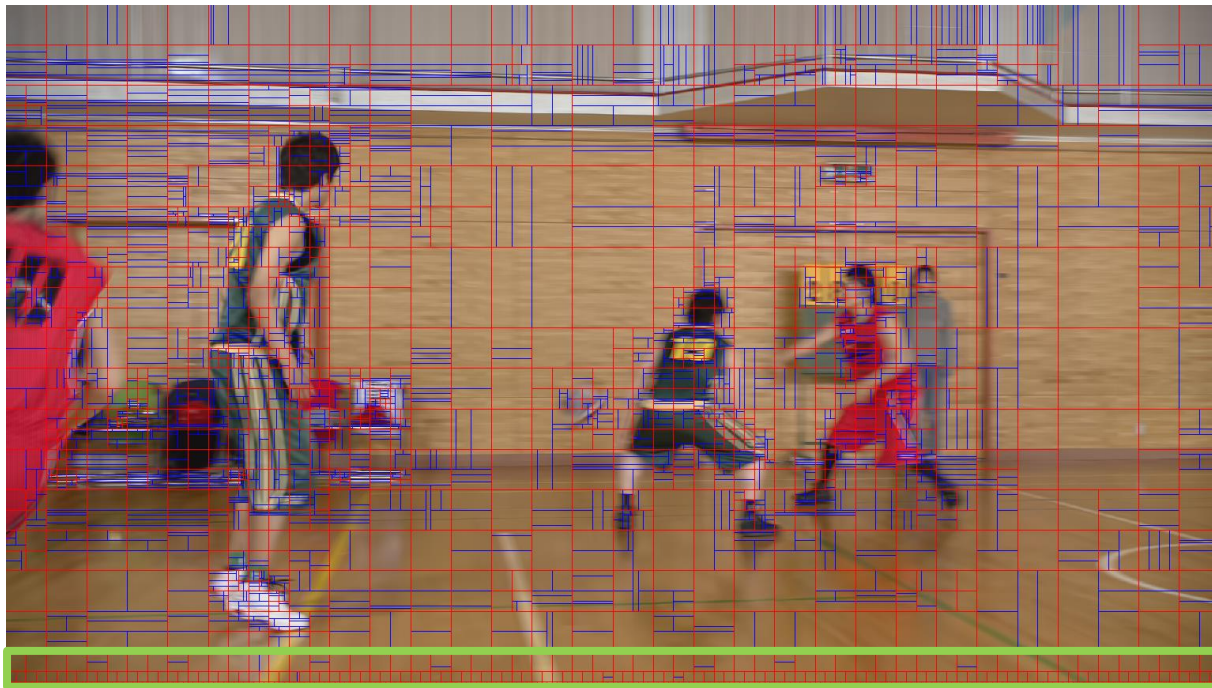
Non-redundancy policy between various successions of splits

- Anti-emulation policy to make each topology reachable through at most one sequence of quad-tree/binary/triple splits



Block splitting on the picture borders

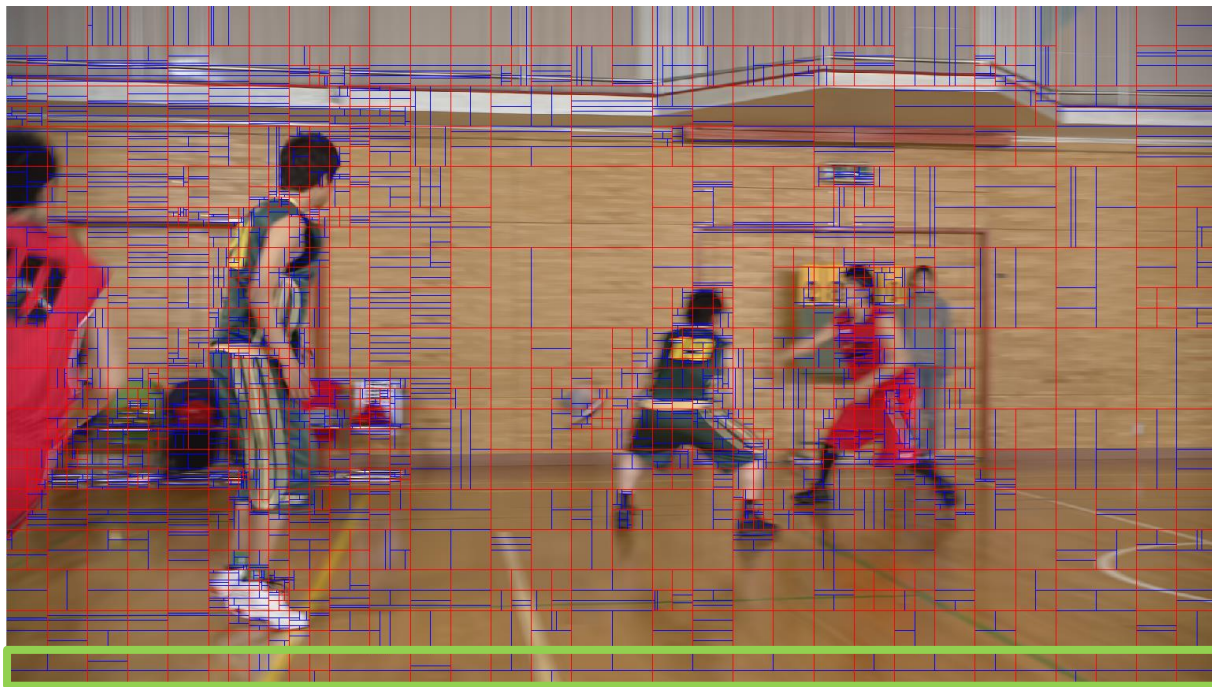
- Gain in CfP RA test set (tool off/tool on): -1.4%/-0.9% (Y)



HEVC-like block splitting
on picture border

Block splitting on the picture borders

- Gain in CfP RA test set (tool off/tool on): -1.4%/-0.9% (Y)



Proposed: RD competition between QT/Binary split successions, matching the picture border

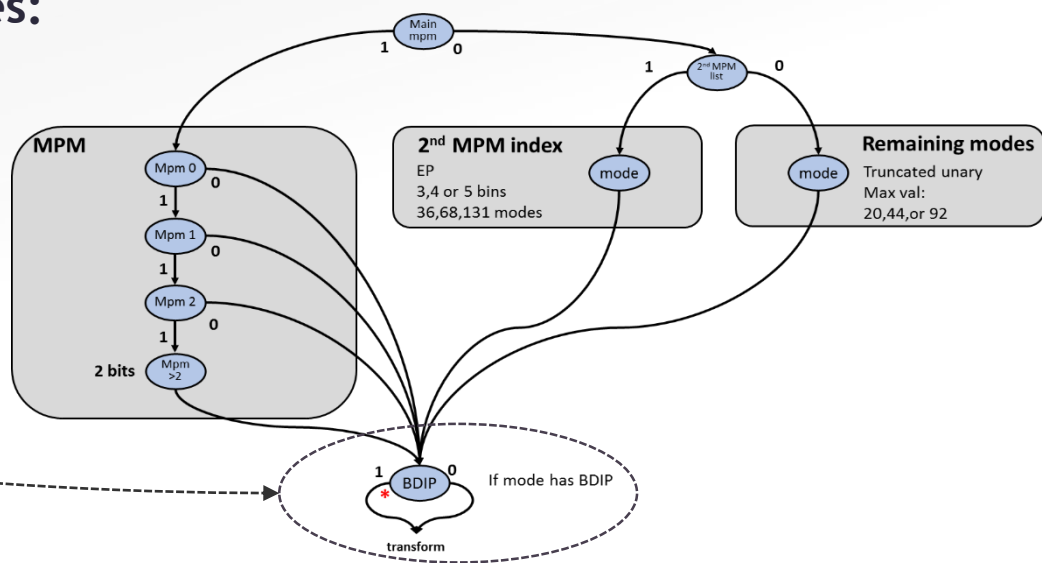
Two intra coding tools are introduced:

- Multi-Reference Intra Prediction
- Bi-directional Intra Prediction

Overall tool off/on performances:

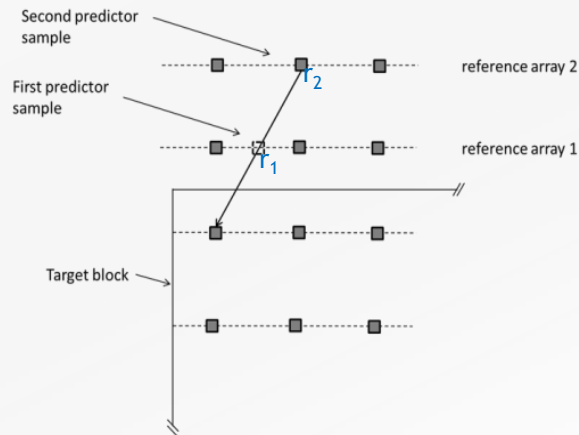
- gain: -0.3%/-0.4% ;
- enc time: 104%/103%;
- dec time: 100%/103%

Intra mode signaling modified



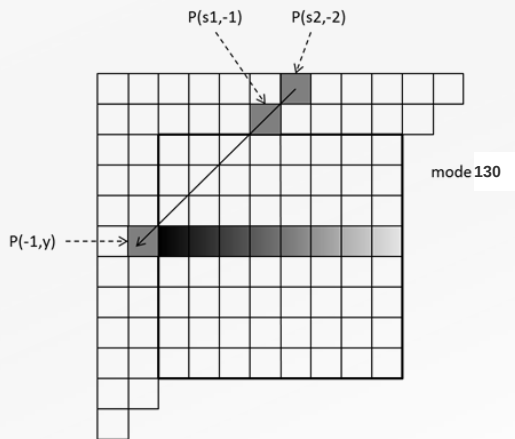
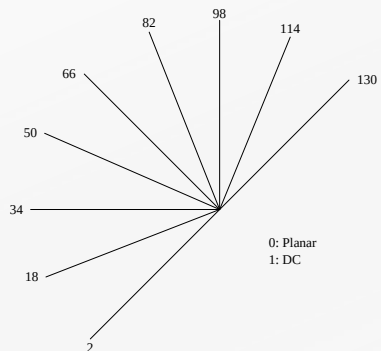
Multi-Reference Intra Prediction (MRIP)

- Tool on/off: -0.2%/-0.2% gain; 102%/109% enc time; 100%/103% dec time



- Principle:
 - 2 reference layers
 - Weighted average of predictions: $pred = (3 \times r_1 + r_2 + 2) \gg 1$
 - Used for non-DC and non-Planar luma Blocks, and for non-LM chroma blocks
- RDO: no additional RD cost computations

Multi-Reference Boundary Filtering: Post-processing step after the prediction to reduce discontinuity



- Intra modes DC, Planar, 98 and 34: same boundary filter as in JEM
- Example for modes 101 to 130

$$P(x, y) = \text{Clip} \left(P(x, y) + \frac{1}{2^x} \left(P(-1, y) - \left(\frac{3}{4} P(s1, -1) + \frac{1}{4} P(s2, -2) \right) \right) \right)$$

Bi-directional Intra Prediction (BDIP)

- Signaled on the CU-level: *BDIPflag*
- Used for luma blocks, intra modes 2-17 and 114-129
- Principle (case of a positive vertical direction):

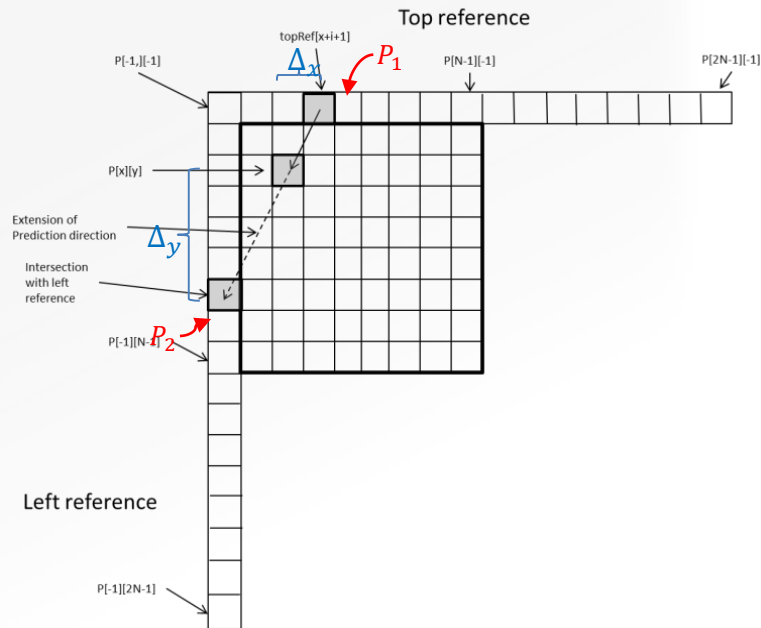
- $\hat{P} = P_1 + \Delta P$

- Prediction update term: $\Delta P = \frac{(P_2 - P_1) \times \Delta_x}{((1+x) \ll 5) + \Delta_x}$

- Interaction between BDIP and MRIP in luma blocks:
 - If *BDIPflag* is true, then BDIP is on, MRIP is off.
 - Else if BDIP off and intra mode is angular, MRIP is on.
 - Else MRIP is off.

- Encoder side:

- Choice between uni-ref BDIP or multi-ref uni-directional prediction: only in fast intra pass
- Full RDO decision between pre-selected tuples (Intra Mode, BDIP flag)



Motion compensation tools: 4 categories

Overall impact on the codec (tool off/on):

- -0.6%/-0.7% gain in luma
- 115%/123% enc. time
- 107%/110% dec. time

Extended template matching

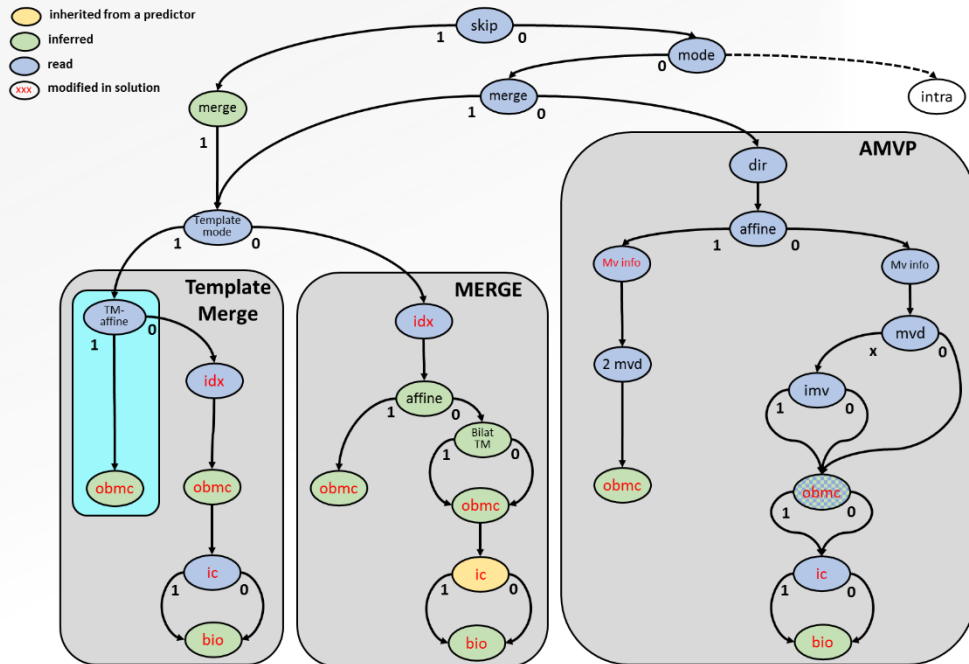
Extended affine motion compensation

- More Affine Candidates
- Template affine prediction mode

Generalized OBMC

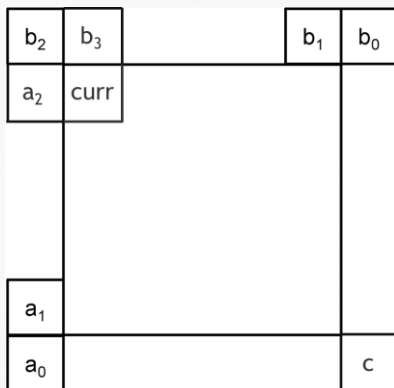
- Design cleaning and simplification

Bi-directional illumination compensation (IC)



Extended Template Matching

■ Tool off/on: Dec.time 97%/93%



	JEM	Proposed
Motion information predictor composition	- inherited reference picture index - inherited associated MV	- inherited reference picture index - inherited associated MV
Candidate list	Up to 13-15 candidates, among: - AMVP candidates if current CU is in AMVP mode - merge candidates - template matching unilateral candidates - spatial neighboring MVs in $\{b_1, a_1\}$	Template matching AMVP Up to 2 candidates, among: - AMVP candidates - merge candidates (non-affine) - spatial neighboring MVs in $\{b_1, a_1\}$
	For sub-blocks, up to 15 candidates, among: - the best candidate of the corresponding entire CU - spatial neighboring motion vectors in $\{b_1, a_1, b_2, b_0\}$ - scaled versions of co-located motion vectors - up to 4 ATMVP candidates - up to 4 STMVP candidates	Template matching merge Up to 4 candidates, among: - merge candidates (non-affine) No more sub-blocks
Signaling	Template Merge mode (1 bin) Residual is coded if mode $\neq$ skip	Template Merge mode (2 bins) Residual is coded if mode $\neq$ skip

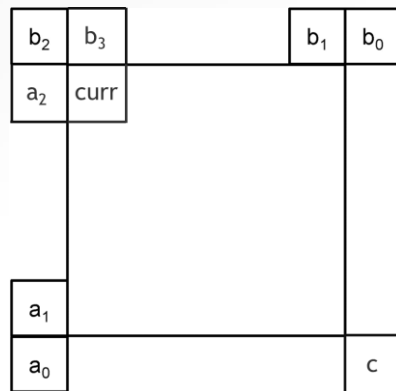
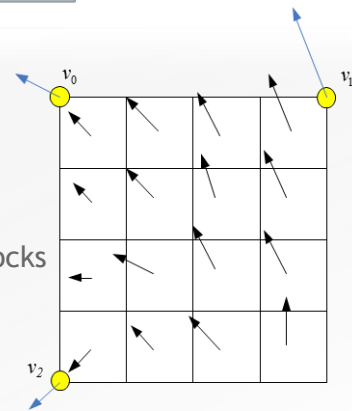
■ Bilateral Motion Vector refinement performed similar as in Template Matching mode

■ Square search along the direction that gave the best



Extended Affine Motion Compensation

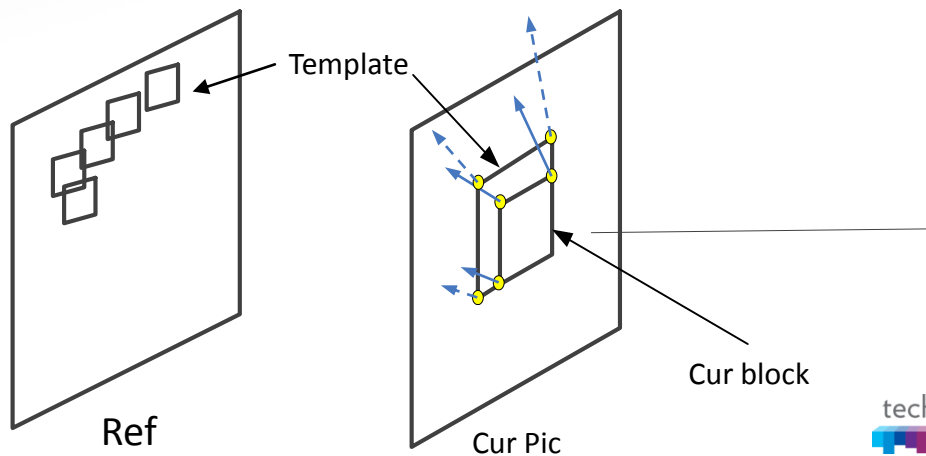
- Same Affine motion estimation and compensation as in JVET-J0021
- Decoupling the motion model from stored MVs on 4x4 block level
 - JEM: affine model seeds stored in the top-left, top-right and bottom-left 4x4 sub-blocks
 - In our proposal: seeds stored separately as a motion info attached to the whole CU
 - Allows using affine mode on CU of size 4 in width or height
- Additional merge affine candidates
 - $\{a_1, b_1, b_0, a_0, b_2, a_2, b_3\}$ in this contribution.
 - temporal affine positions in $\{\text{curr}, c\}$
- Temporal Affine Search/Derivation
 - Search affine block at position $\{\text{curr}, c\}$ in ref pics
 - If found:
 - Retrieve its Affine Model
 - Inherit the candidate ref pic
 - Scale affine model seeds scaling according to POC distances between pictures



Template Affine Prediction Mode (used in Affine AMVP and template merge modes)

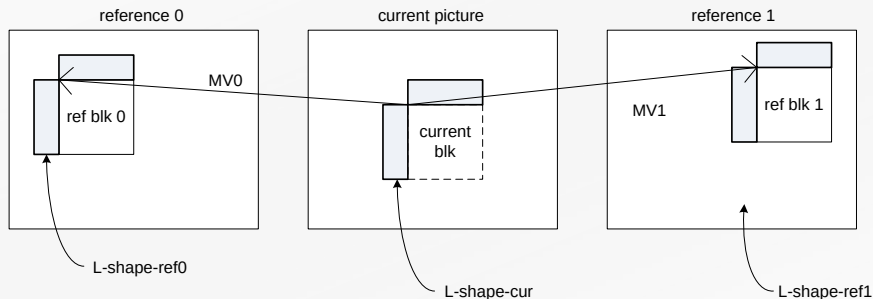
■ Goal: automatically find an affine motion model for the current CU and refine it

1. For each ref pic list, choose a starting point affine model:
 - First spatial affine neighbor in $\{a1, b1, b0, a0, b2, a2, b3\}$ if available
 - MV issued from a template matching search otherwise
2. Derive an Affine model for the considered 4-pixels thick template
3. Gradient-based search to minimize the template SAD cost
 - Refinement of each seed, 5 iterations at most



Bi-directional Illumination Compensation

■ IC:



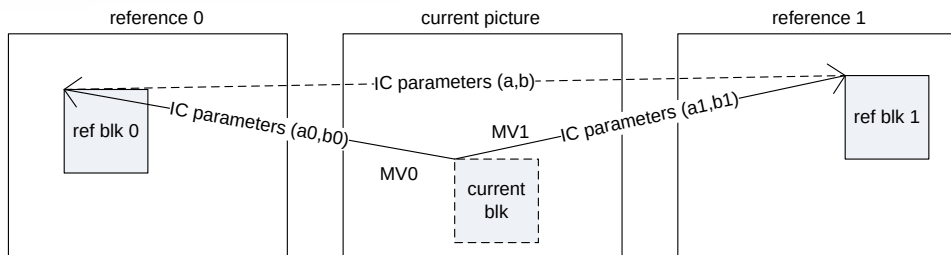
$$predCurr = \frac{1}{2}(a_0.recRef_0 + b_0) + \frac{1}{2}(a_1.recRef_1 + b_1)$$

■ Proposed Bi-directional IC (Gain: -0.2%; Enc.Time: 102%):

1. Estimate IC params between ref blocks $\rightarrow (a, b)$
2. Derive IC parameters $(a_i, b_i), i = 0, 1$ between current block and ref blocks:
3. For block size higher than 8, choice between uni-dir and bi-dir IC

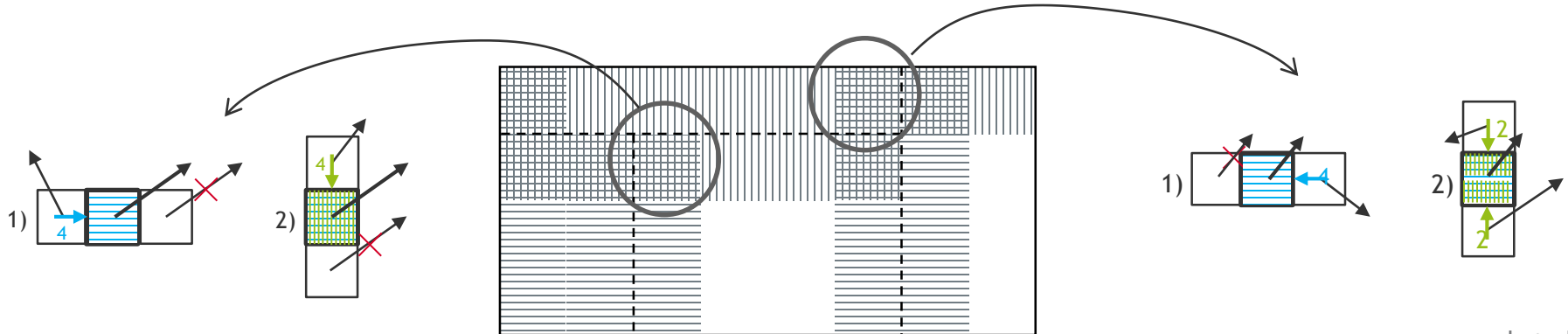
$$\begin{cases} a_0 = a \cdot \alpha + (1 - \alpha) \\ b_0 = \alpha \cdot b \\ a_1 = \alpha + (1 - \alpha)/a \\ b_1 = -b(1 - \alpha)/a \end{cases}$$

$$\text{where: } \alpha = \frac{poc_{cur} - poc_0}{poc_1 - poc_0}$$



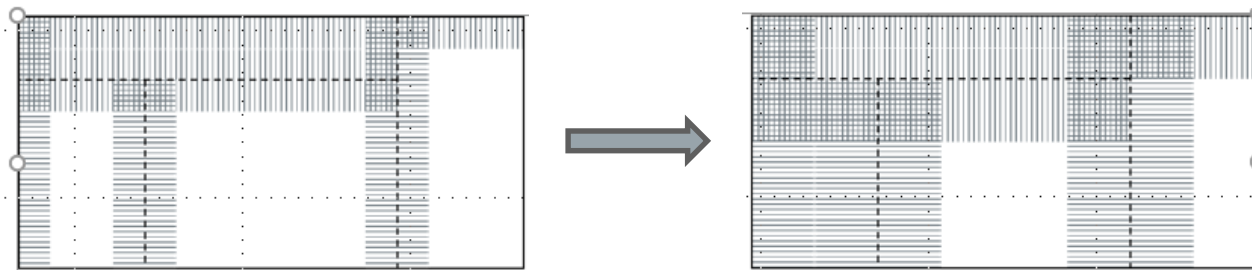
Generalized OBMC

- Decision to filter 2 or 4 lines/columns taken on a 4x4 sub-block basis
 - According to motion vectors located at opposite 4x4 sub-block borders
 - 4x4 sub-block borders considered by pair (left, right) then (top, bottom)
 - For each pair:
 - if 2 neighboring MVs available and different from current MV, then apply OBMC on 2 lines/cols
 - If only 1 neighbor MV available or only 1 different from current MV, apply OBMC on 4 lines/cols on concerned border
 - If no neighboring MV available or both neighbor MVs equal to current MV, do not apply OBMC in concerned orientation



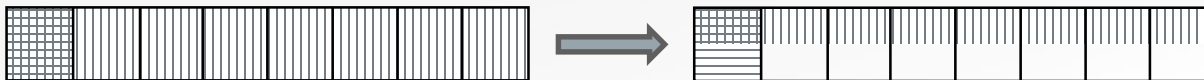
Generalized OBMC

- Decision to filter 2 or 4 lines/columns taken on a 4x4 sub-block basis
 - According to motion vectors located at opposite 4x4 sub-block borders
 - 4x4 sub-block borders considered by pair (left, right) then (top, bottom)
 - For each pair:
 - if 2 neighboring MVs available and different from current MV, then apply OBMC on 2 lines/cols
 - If only 1 neighbor MV available or only 1 different from current MV, apply OBMC on 4 lines/cols on concerned border
 - If no neighboring MV available or both neighbor MVs equal to current MV, do not apply OBMC in concerned orientation
- Advantage:
 - Allow different numbers of filtered samples for horizontal and vertical directions
 - Better adaptation to motion transitions within a CU. Example of a merge ATMVP CU:

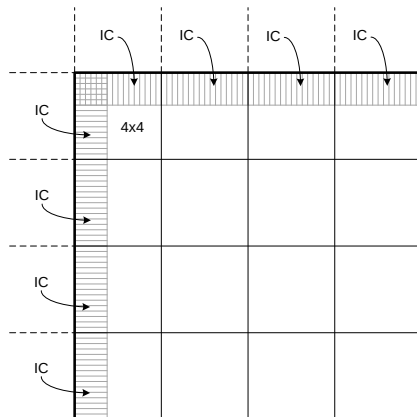


Generalized OBMC

- Further adaptation for thin CUs of size smaller than 8 in width or height
 - Only 2 lines/cols filtered in concerned orientation



- OBMC design simplification
 - IC parameters computation for each OBMC 4xS band is removed
 - IC parameters inherited from considered neighboring 4x4 sub-blocks instead



Transform set and Primary/Secondary transforms interaction

- Reduced pool of primary transforms in EMT
- Interaction constraints on EMT/NSST Transforms
- Overall impact on the codec (tool on/off):
 - 0.1%/0.1% loss in luma
 - 99%/96% enc. time
 - 100%/100% dec. time
 - memory reduction

Transforms

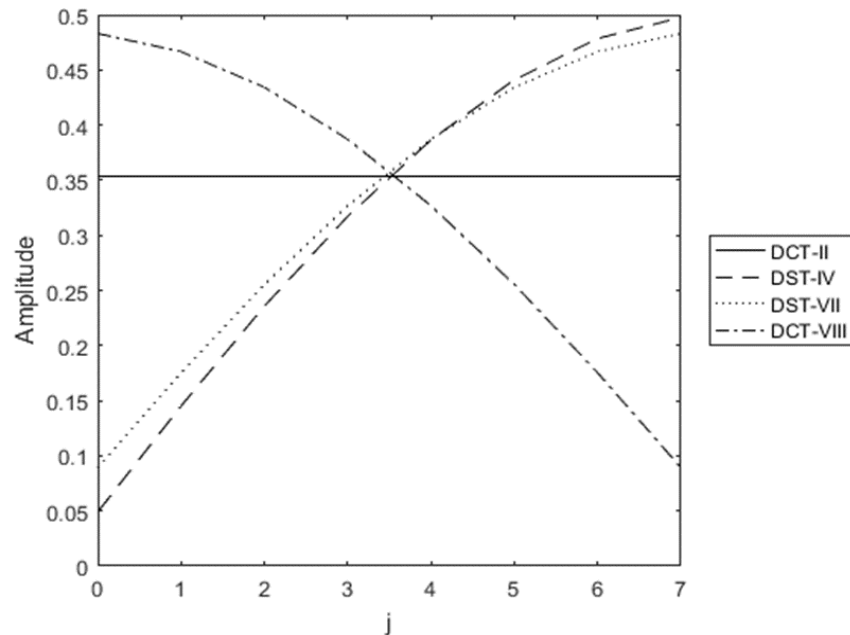
Reduced pool of separable primary transforms

- Rationale: keep the transforms of interest for compression efficiency and practical for hardware implementation
- Observations:
 - DCT-V very close to DCT-II
 - DST-I brings very small coding efficiency improvement

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-V	$T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N-1}} \cdot \cos\left(\frac{2\pi \cdot i \cdot j}{2N-1}\right),$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}, \omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-I	$T_i(j) = \sqrt{\frac{2}{N+1}} \cdot \sin\left(\frac{\pi \cdot (i+1) \cdot (j+1)}{N+1}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

Reduced pool of separable primary transforms

- **Rationale:** keep the transforms of interest for compression efficiency and practical for hardware implementation
- **Observations:**
 - DCT-V very close to DCT-II
 - DST-I brings very small coding efficiency improvement
- **Proposed Reduced Transform Set:**
 - One Discrete Trigonometric Transform (DTT) with constant lowest frequency basis function: **DCT-II**
 - One DTT with monotonically increasing lowest frequency basis function: **DST-VII** (KLT in the direction of intra pred)
 - One DTT with monotonically decreasing lowest frequency basis function: **DCT-VIII**
 - **DST-IV** is used since it brings variation to DST-VII



Reduced pool of separable primary transforms

- **Rationale:** keep the transforms of interest for compression efficiency and practical for hardware implementation
- **Observations:**
 - DCT-V very close to DCT-II
 - DST-I brings very small coding efficiency improvement
- **Proposed Reduced Transform Set:**
 - One Discrete Trigonometric Transform (DTT) with constant lowest frequency basis function: **DCT-II**
 - One DTT with monotonically increasing lowest frequency basis function: **DST-VII** (KLT in the direction of intra pred)
 - One DTT with monotonically decreasing lowest frequency basis function: **DCT-VIII**
 - **DST-IV** is used since it brings variation to DST-VII

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2}\right)$
DST-IV	$T_i(j) = \sqrt{\frac{2}{N}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

Transforms

Reduced pool of separable primary transforms

■ Further observation:

$$T_{DST-VII} = R \times T_{DCT-VIII} \times S$$

■ where:

$$R(i, j) = \begin{cases} 1 & \text{if } i = N - 1 - j \\ 0 & \text{else} \end{cases} \quad S(i, j) = \begin{cases} (-1)^j & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

■ Proposal:

- DST-VII is replaced by the mirror of DCT-VIII → memory saving
- DCT-VIII is kept for reduced memory consumption because it is orthogonal and symmetric → self-inverse:

$$T_{DCT-VIII}^{-1} = T_{DCT-VIII}^t = T_{DCT-VIII}$$

■ Result:

- Less than 0.1% loss in BD-rate
- Memory saving: 50%

Transform Type	Basis function $T_i(j)$, $i, j=0, 1, \dots, N-1$
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j + 1)}{2N}\right)$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \cos\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N + 2}\right)$
DST-IV	$T_i(j) = \sqrt{\frac{2}{N}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (2j + 1)}{4N}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N + 1}} \cdot \sin\left(\frac{\pi \cdot (2i + 1) \cdot (j + 1)}{2N + 1}\right)$

	TrIdx	0	1	2	3
predModIdx					
0		DST4,DST4	DST7,DST7	DST4,DCT8	DCT8,DST4
1		DST7,DST7	DCT2,DCT2	DCT2,DST7	DCT2,DCT8
2		DST7,DST7	DST7,DCT8	DCT8,DST7	DCT2,DST7
3		DST4,DST4	DST4,DCT2	DCT8,DST4	DCT2,DST7
4		DST4,DST7	DST7,DCT2	DCT8,DST7	DCT2,DST7
5		DST7,DST7	DST7,DCT2	DCT8,DST7	DCT2,DST7
6		DST7,DST7	DST7,DCT2	DCT2,DST7	DCT2,DST7

Transforms

Interaction Between Primary Transform, Secondary Transform and Transform Skip

■ Initially:

- secondary (NSST) + primary (EMT) inverse transforms
- Or: transform skip (Tskip)

■ Observation:

- NSST and EMT give compression gain when used independently, but gain is much lower when used together
- EMT+NSST imply more than 20 candidates for primary/secondary transforms

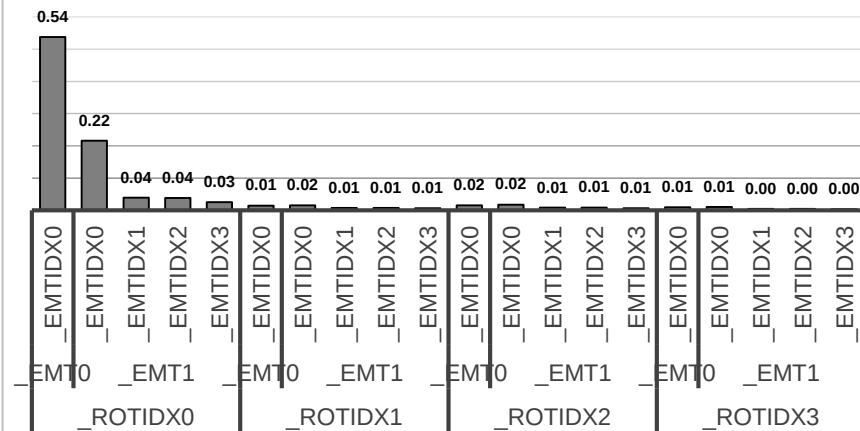
■ Proposal

- **NSST activated when EMT CU flag is off or EMT TrIdx=0**

■ Result:

- No loss in BD-rate (RA), Encoding Time Reduction by 10/15%

EMT-NSST combination usage histogram



SAO Palette

- Tool off/on: -0.2%/-0.3% Y, 100% enc. time, 101%/108% dec. time
- HEVC: merge mode uses top or left SAO params

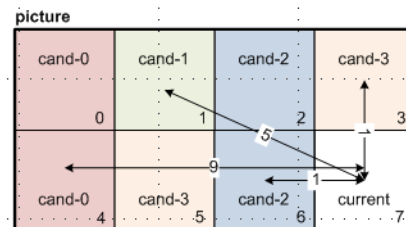
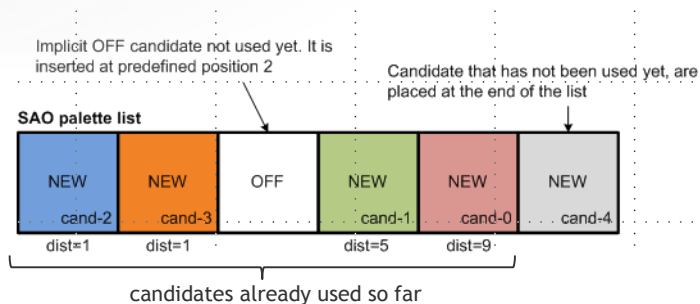
NEW 0	MERGE cand-0 1	NEW 2	MERGE cand-1 3
MERGE cand-0 4	MERGE cand-0 5	NEW 6	MERGE cand-2 7

- SAO Palette proposal:

- Decode all the SAO parameters (NEW) first, for a slice.
- Then blocks may inherit SAO parameters from all the other blocks.
- For each block, syntax element *sao_palette_idx* indicates its SAO parameters
- The palette is sorted with SAO parameters of closest neighbors on top of the list

NEW cand-0	NEW cand-1	NEW cand-2
---------------	---------------	---------------

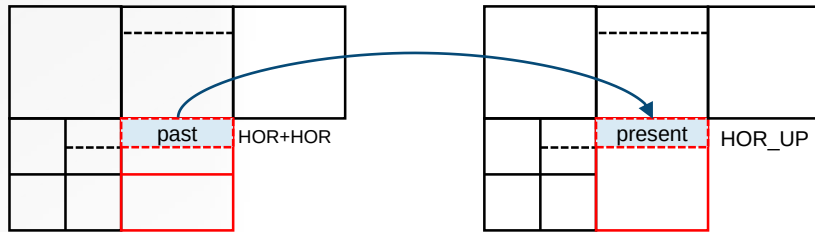
MERGE cand-0 0	MERGE cand-0 1	MERGE cand-1 2	MERGE cand-1 3
MERGE cand-0 4	MERGE cand-0 5	MERGE cand-2 6	MERGE cand-2 7



Fast coding methods

■ Caching mechanism to re-use already optimized coding parameters

- Full re-use (same causal environment)
- Partial re-use when different causality (reduced RDO based on stored data)



■ RD-cost based heuristics in B slices

- If($\text{RDCost}(\text{HOR_SPLIT}) > \alpha \times \text{RDCost}(\text{NO_SPLIT})$) then disable asymmetric horizontal
- If($\text{RDCost}(\text{VER_SPLIT}) > \alpha \times \text{RDCost}(\text{NO_SPLIT})$) then disable asymmetric vertical
- If($\text{RDCost}(\text{HOR_SPLIT}) > \beta \times \text{RDCost}(\text{VER_SPLIT})$) then disable asymmetric horizontal
- If($\text{RDCost}(\text{VER_SPLIT}) > \beta \times \text{RDCost}(\text{HOR_SPLIT})$) then disable asymmetric vertical

■ Asymmetric partitioning ratio (1/3,2/3) is deactivated.

■ Block size 48 only supported in luma in I slices.

■ CNN-based split mode prediction in Intra Slices

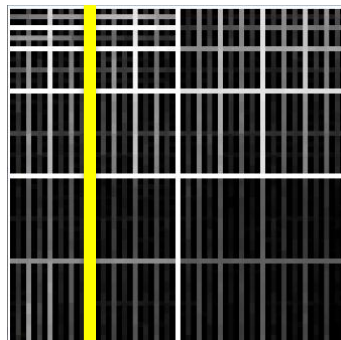
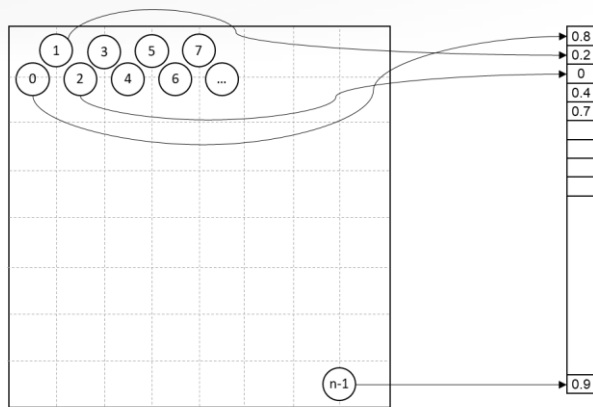
Fast coding methods

■ CNN-based split mode prediction in Intra slices (detailed in JVET-J0034)

- Goal: leverage the gains of QTBT and ABT, while keeping the combinatory reasonable
- Principle: given a block and its QT and BT depths, select a limited set of split modes to be RD-evaluated

■ Process

1. **Texture Analysis by the CNN of a root block** (64x64 block, plus its causal left and top neighbor boundaries) → vector of 480 probabilities associated to each elementary boundary between 4x4 sub-blocks
2. **For each candidate split boundary, compute its probability** as the integral of elementary probabilities over the candidate split boundary → select a set of split modes given their probabilities
3. **Reduce the set of split modes to be RD-checked**, through the non-redundancy policy between split modes, and the encoder speed-ups



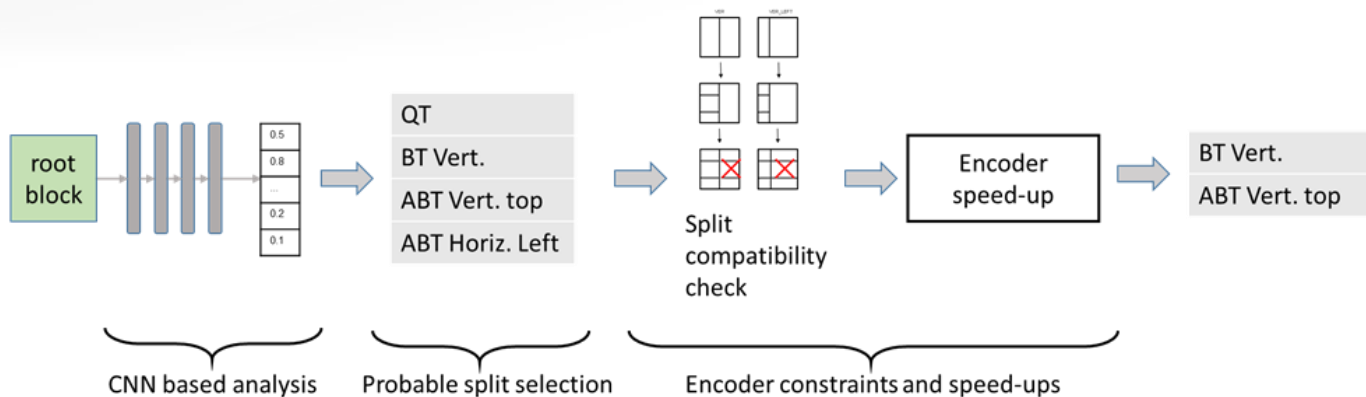
Fast coding methods

■ CNN-based split mode prediction in Intra slices (detailed in JVET-J0034)

- Goal: leverage the gains of QTBT and ABT, while keeping the combinatory reasonable
- Principle: given a block and its QT and BT depths, select a limited set of split modes to be RD-evaluated

■ Process

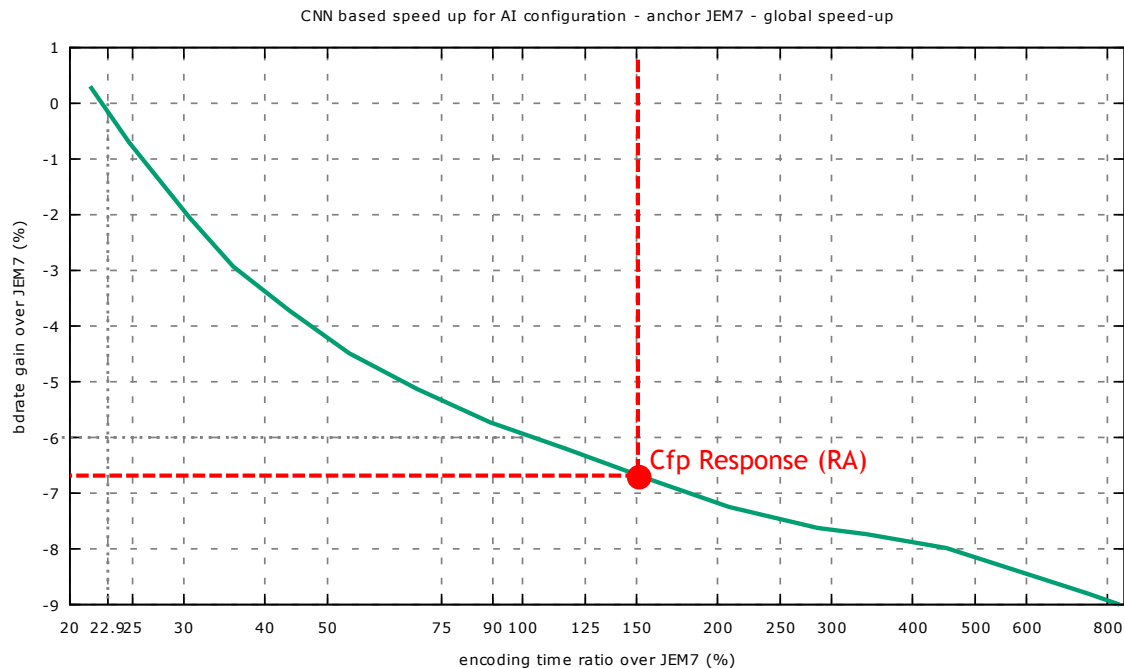
1. **Texture Analysis by the CNN of a root block** (64x64 block, plus its causal left and top neighbor boundaries) → vector of 480 probabilities associated to each elementary boundary between 4x4 sub-blocks
2. **For each candidate split boundary, compute its probability** as the integral of elementary probabilities over the candidate split boundary → select a set of split modes given their probabilities
3. **Reduce the set of split modes to be RD-checked**, through the non-redundancy policy between split modes, and the encoder speed-ups



All-Intra codec behavior

■ A coder control parameter “speed-control” jointly selects:

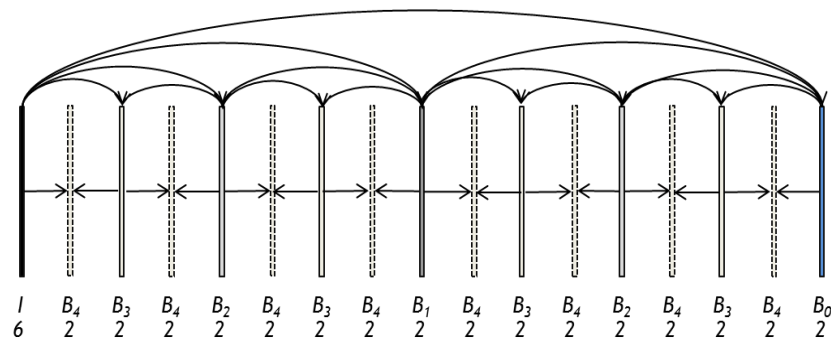
- the structure of the tree used for the luma and chroma: maximum and minimum QT depth and BT/ABT depth
- the probability thresholds, used to select split mode candidates



Various Coding Tree / ABT configurations

Medium complexity (CfP bit-streams)

- Combinatory allocation adaptive to slice type/Temporal ID:
 - BT max depth=6 for Intra Slices (feasible through CNN based split prediction)
- Max QT depth varies by slice type, temporal level and component
- Max BT depth varies by slice type, T-layer and QT depth level.
- On-the-fly evolution of the coding tree by temporal layer



Slice Type	T-Id	Min/Max QT depth	Component type	Max BT depth for each QT depth = 0...5					
				QT Depth 0 SizeY 256	QT Depth 1 SizeY 128	QT Depth 2 SizeY 64	QT Depth 3 SizeY 32	QT Depth 4 SizeY 16	QT Depth 5 SizeY 8
I	0	0/3	Luma	0	0	2	6		
			Chroma	0	2	6	0		
B	0	0/5	Luma/Chroma	0	3	3	3	3	3
B	1	0/5	Luma/Chroma	0	2	2	2	2	2
B	2	0/5	Luma/Chroma	0	2	2	2	2	2
B	3	0/5	Luma/Chroma	0	2	2	2	2	2
B	4	0/5	Luma/Chroma	0	2	2	2	2	2

Medium complexity Objective Results

■ RA

- Over JEM7: -13.7% gain, enc. time 90%, dec. time 82%
- Over HM: -41.9% gain, enc. time 728%, dec. time 582%

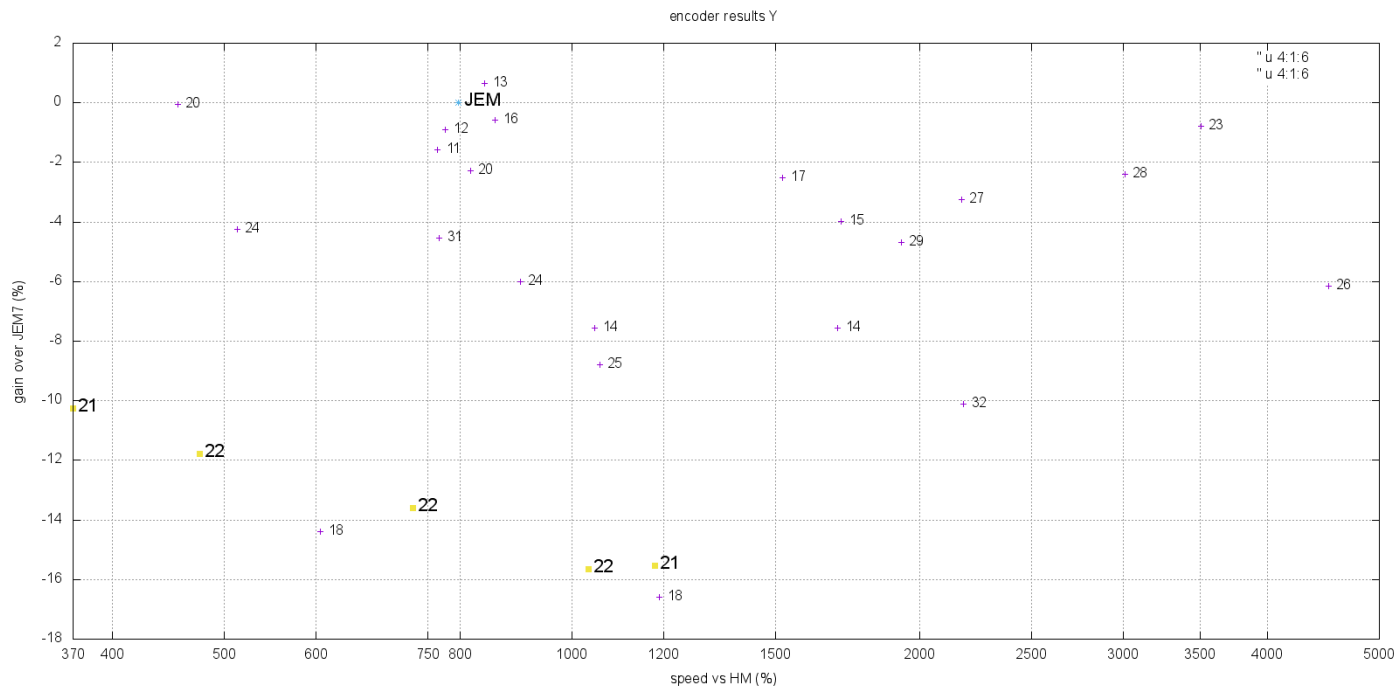
■ LDB

- Over JEM7: -12.7% gain, enc. time 119%, dec. time 89%
- Over HM: -33.8% gain, enc. time 1035%, dec. time 524%

SDR	Over HM anchors					Over JEM anchors				
	BD-rate PSNR			Time variation		BD-rate PSNR			Time variation	
	Y	U	V	Enc	Dec	Y	U	V	Enc	Dec
CS1	-41.9%	-44.8%	-46.1%	728%	582%	-13.6%	-3.8%	-5.6%	90%	82%
CS2	-33.8%	-51.6%	-52.5%	1035%	524%	-12.7%	-16.7%	-18.8%	119%	89%

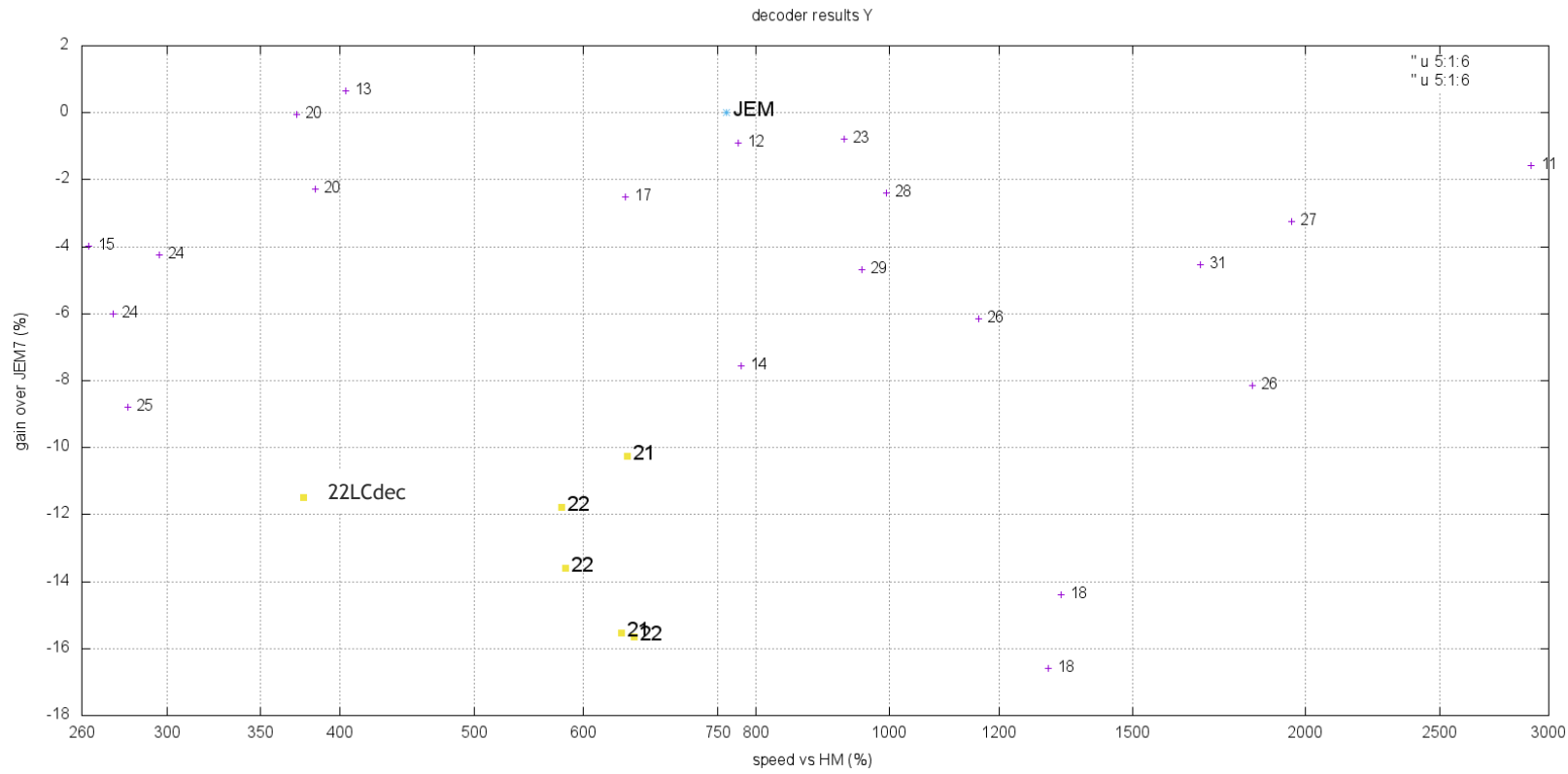
Additional Data provided in JVET-J0022 (QTBT+ABT + J0022 tools)

- **High Efficiency** encoder: **-15.6%** over JEM7, enc. time **127%**
- **Low Complexity** encoder: **-11.8%** over JEM7, enc. time **59%**
- **[last minute results]** Low Complexity **decoder**: **-11.5%** over JEM7, enc. Time 77%, dec.time **53%**



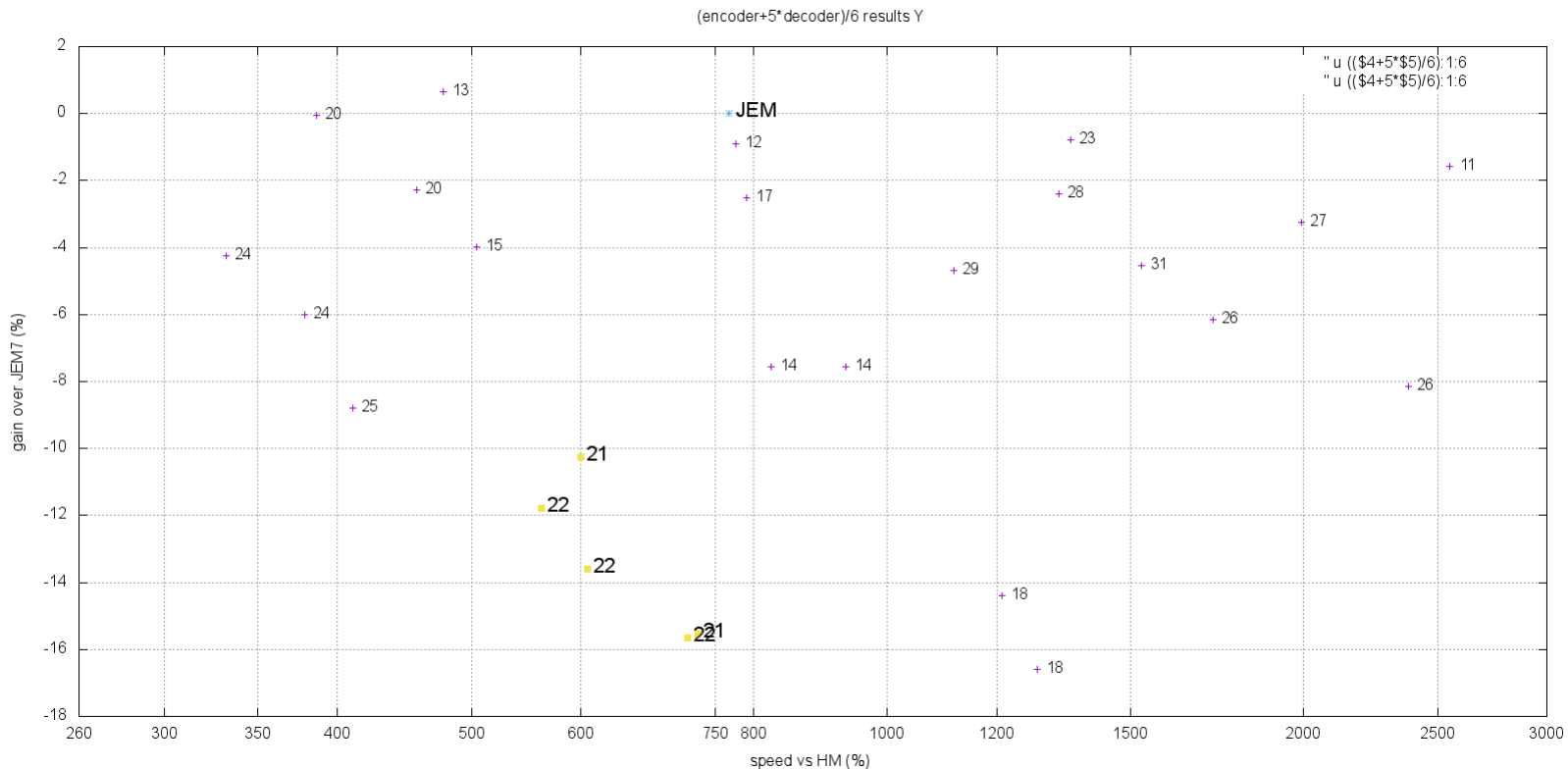
Overview of JVET-J0022 trade-offs

■ BD-rate-over-JEM7 = f (dec.time over HM)



Overview of JVET-J0022 trade-offs

■ BD-rate-over-JEM7 = $f(\text{enc.time} + 5 \times \text{dec.time})$



Tools description - VR category



Adaptive Quantization for ERP

- Equirectangular projection de facto broadly used
- Issue: bad sampling w.r.t actual rendered frames
- Adaptive QP proposed to balance bit budget
- QP weighting follows WS-PSNR metric:

$$QP(x, y) = \min(QP_{base} - 3 * \log_2(w(y)), QP_{max})$$

$$w(y) = \frac{\cos(\pi y) h}{\sum_h w(y)}$$

with $QP_{max}=51$ and h the image height.

- CTU-based granularity in CfP response

	E2E WS-PSNR			E2E SPSNR-NN		
	BD-rate(Y)	BD-rate(U)	BD-rate(V)	BD-rate(Y)	BD-rate(U)	BD-rate(V)
Balboa	-2.2%	-2.8%	-3.3%	-2.1%	-2.8%	-3.2%
Chairlift	-5.7%	-3.1%	-3.8%	-5.8%	-3.1%	-3.7%
KiteFlite	-3.3%	-5.1%	-8.8%	-3.3%	-5.0%	-8.8%
Harbor	-1.7%	-4.3%	-4.1%	-1.7%	-4.2%	-4.0%
Trolley	-2.0%	-0.1%	-3.1%	-2.0%	0.0%	-3.1%
Average	-3.0%	-3.1%	-4.6%	-3.0%	-3.0%	-4.6%

Performance, tool on/off within

J0022 CfP response

360 video specific developments

QP=34

QP=29

QP=27

QP=29

QP=34

-2 8 18

— pixel-based weighting
— CTU-based delta QP

360 video specific developments

Over HM16.16

	E2E WS-PSNR			E2E SPSNR-NN		
	BD-rate(Y)	BD-rate(U)	BD-rate(V)	BD-rate(Y)	BD-rate(U)	BD-rate(V)
Balboa	-46.2%	-54.2%	-58.9%	-46.1%	-54.2%	-58.9%
Chairlift	-44.1%	-58.7%	-55.2%	-44.1%	-58.8%	-55.1%
KiteFlite	-23.7%	-52.2%	-57.1%	-23.7%	-52.2%	-57.1%
Harbor	-25.4%	-54.7%	-54.5%	-25.4%	-54.7%	-54.4%
Trolley	-21.2%	-38.2%	-42.4%	-21.2%	-38.3%	-42.3%
Average	-32.1%	-51.6%	-53.6%	-32.1%	-51.6%	-53.6%
Enc Time[%]	579%					
Dec Time[%]	519%					

Over JEM7.0

	E2E WS-PSNR			E2E SPSNR-NN		
	BD-rate(Y)	BD-rate(U)	BD-rate(V)	BD-rate(Y)	BD-rate(U)	BD-rate(V)
Balboa	-17.2%	-20.2%	-22.3%	-17.1%	-20.2%	-22.3%
Chairlift	-19.8%	-16.4%	-18.4%	-19.9%	-16.5%	-18.4%
KiteFlite	-7.7%	-8.4%	-9.4%	-7.6%	-8.4%	-9.4%
Harbor	-6.9%	-16.1%	-14.7%	-6.9%	-16.1%	-14.7%
Trolley	-6.7%	-4.3%	-5.9%	-6.7%	-4.3%	-5.8%
Average	-11.7%	-13.1%	-14.1%	-11.7%	-13.1%	-14.1%
Enc Time[%]	97%					
Dec Time[%]	102%					

Tools description - HDR category



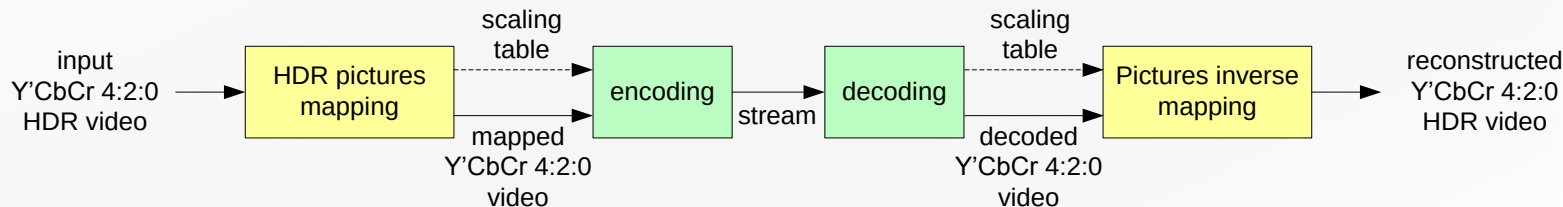
2 out-of-loop tools

- Dynamic range adaptation as pre-encoding / post-decoding processing
- Post-decoding refinement

Basics:

- Signal reshaping in the Y'CbCr 4:2:0 domain
- Using samples located at the same relative picture location (no chroma upsampling, no conversion to linear-light)

Reshaping with luma-based cross-component scaling



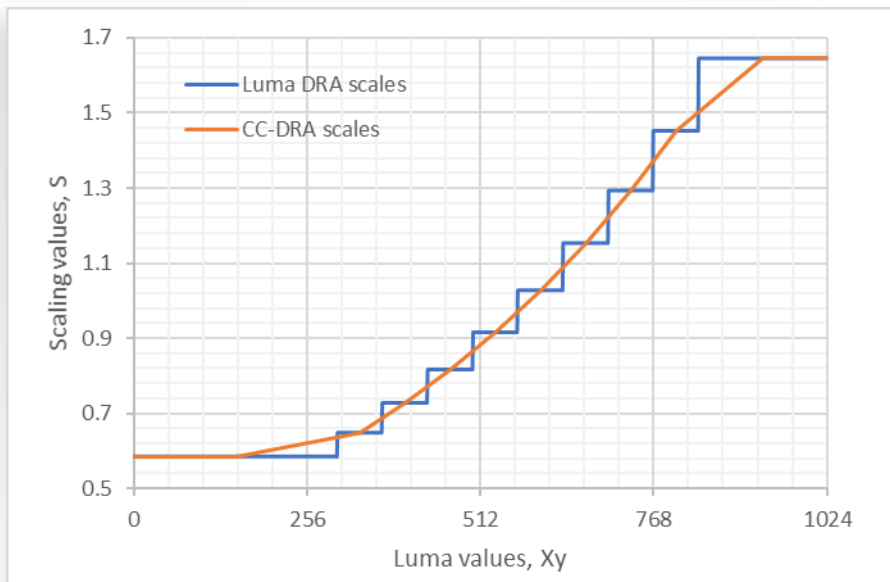
1 single scaling table used for both mapping Y' and scaling Cb and Cr

Reproduces behaviour of luma-based QP adaptation with a pixel-level accuracy

- In luma-based QP adaptation, the same luma-dependent dQP is used for scaling the luma and the chroma; this results in a luma-dependent cross-component scaling.
- In the proposed reshaping, the luma-dependent cross-component scaling is achieved at pixel-level.

Dynamic range adaptation

Range	$Y_0 - Y_1$	$Y_1 - Y_2$	...	$Y_{N-2} - Y_{N-1}$
scaling	Sc_0	Sc_1	...	Sc_{N-1}



Luma mapping

$$\blacksquare \text{map}(Y) = \text{map}(Y_i) + Sc_i * (Y - Y_i)$$

Luma-based chroma scaling

$$\blacksquare C_{\text{map}} = \text{offset} + Sc_i * (C_{\text{in}} - \text{offset})$$

$$\blacksquare C = C_b, C_r$$

$$\blacksquare i \text{ interval of co-located luma sample : } Y \text{ in } [Y_i, Y_{i+1}]$$

Aims at compensating distortion resulting from the compression

The inverse mapped signal (Y_{IM}, U_{IM}, V_{IM}) is refined as follows:

■ For luma component, intra-component scaling:

■ $Y_{refine} = \text{Min}(2^{B_{lum}} - 1, \text{LUT}_{refineY}[Y_{IM}] * Y_{IM})$ can be achieved using a direct LUT

■ For chroma components, luma-dependent scaling:

■ $U_{refine} = \text{Max}(0, \text{Min}(2^{B_{chr}} - 1, 2^{B_{chr}-1} + \text{LUT}_{refineU}[Y_{IMcoloc}] * (U_{IM} - 2^{B_{chr}-1})))$

■ $V_{refine} = \text{Max}(0, \text{Min}(2^{B_{chr}} - 1, 2^{B_{chr}-1} + \text{LUT}_{refineV}[Y_{IMcoloc}] * (V_{IM} - 2^{B_{chr}-1})))$

Refinement tables coded as piece-wise linear models (slice level)

PQ : DRA based on same dQP tables as the anchors (fixed for all sequences)

HLG : no DRA applied

		Over JEM7.0							
		BD-rate							
		DE100	PSNRL100	wPsnrY	wPsnrU	wPsnrV	psnrY	psnrU	psnrV
HDR-A	DayStreet	-19.0%	-10.0%	-9.0%	-8.4%	-21.3%	-8.7%	-6.1%	-14.4%
	PeopleInShoppingCenter	-9.2%	-8.3%	-7.9%	-9.8%	-7.0%	-7.8%	-6.4%	-5.8%
	SunsetBeach	-21.5%	-9.3%	-8.2%	4.5%	-31.6%	-7.6%	0.8%	-22.9%
HDR-B	Market	-6.8%	-14.2%	-11.8%	11.0%	49.1%	-5.9%	27.3%	77.1%
	ShowGirl2	-22.3%	-14.0%	-12.0%	-25.1%	-34.3%	-8.9%	-29.5%	-34.4%
	Hurdles	-28.7%	-16.4%	-14.1%	-11.6%	-27.7%	-11.9%	0.7%	-12.2%
	Starting	-13.1%	-18.5%	-14.4%	9.4%	21.4%	-11.7%	19.9%	70.5%
	Cosmos1	-15.4%	-15.4%	-12.1%	39.7%	-21.0%	-11.2%	32.3%	-22.2%
Average HDR-A		-16.6%	-9.2%	-8.4%	-4.6%	-20.0%	-8.0%	-3.9%	-14.4%
Average HDR-B		-17.3%	-15.7%	-12.9%	4.7%	-2.5%	-9.9%	10.1%	15.8%
Average all		-17.0%	-13.3%	-11.2%	1.2%	-9.1%	-9.2%	4.9%	4.5%
Enc Time[%]		96%							
Dec Time[%]		90%							

Per-tool (PQ only)

Tool name	DE100	PSNRL100	wPsnrY	wPsnrU	wPsnrV	Enc	Dec
DRA+refinement	-44.3%	-9.8%	-3.8%	-49.2%	-83.8%	97%	100%
DRA	-42.6%	-9.9%	-3.6%	-45.7%	-81.1%	95%	101%
refinement	-3.6%	0.0%	-0.1%	-8.2%	-13.3%	101%	98%

Conclusion



MTT software model proposed in JVET-J0021/J0022 for SDR, HDR and VR360

Proposed tools in J0022

RA all	“tool-off” test results					“tool-on” test results				
name										
	BD-rate			Runtime		BD-rate			Runtime	
tools	Y	Cb	Cr	Enc	Dec	Y	Cb	Cr	Enc	Dec
TOPOLOGY	-5.0%	-5.6%	-5.5%	185%	98%	-4.3%	-6.4%	-6.6%	201%	101%
ABT(over QTBT)	-3.8%	-4.1%	-4.3%	186%	99%	-3.2%	-5.0%	-5.2%	192%	101%
Pict. border constraints	-1.4%	-2.0%	-1.9%	102%	99%	-0.9%	-1.1%	-1.2%	103%	100%
INTRA	-0.3%	-0.7%	-0.5%	104%	100%	-0.4%	-0.5%	-0.3%	113%	103%
Multiref IP	-0.2%	-0.3%	-0.3%	102%	101%	-0.2%	-0.4%	-0.2%	109%	102%
Bidir IP	-0.2%	-0.3%	0.0%	102%	100%	-0.2%	0.0%	-0.1%	103%	100%
MVCODING+MC	-0.6%	-0.5%	-0.4%	115%	107%	-0.7%	-0.7%	-0.7%	123%	110%
Extended Affine MC	-0.3%	-0.2%	-0.1%	110%	106%	-0.5%	-0.6%	-0.7%	111%	115%
Bidir IC	-0.1%	-0.1%	0.1%	104%	105%	-0.2%	-0.2%	-0.1%	104%	108%
Template Merge	-0.1%	-0.1%	0.2%	102%	97%	0.0%	0.1%	-0.1%	100%	93%
Generalized OBMC	-0.1%	-0.1%	0.1%	102%	99%	0.0%	-0.2%	0.0%	99%	99%
TRANSFORM	0.1%	0.0%	-0.1%	99%	100%	0.1%	-0.3%	-0.2%	96%	100%
Transforms Reduc	0.0%	0.2%	0.2%	100%	100%	0.0%	-0.1%	0.4%	100%	100%
Interaction EMT-NSST	0.1%	-0.1%	0.2%	97%	100%	0.1%	-0.3%	-0.3%	96%	101%
LOOP FILTER	-0.2%	-0.5%	-0.5%	100%	101%	-0.3%	-0.6%	-0.7%	100%	108%
SAO palette	-0.2%	-0.5%	-0.5%	100%	101%	-0.3%	-0.6%	-0.7%	100%	108%

- Most tools focused on coding improvements
- Finer block structure as a core coding tool
- Other tools presenting interesting gain/complexity tradeoffs
- Some tools more focused on complexity reduction

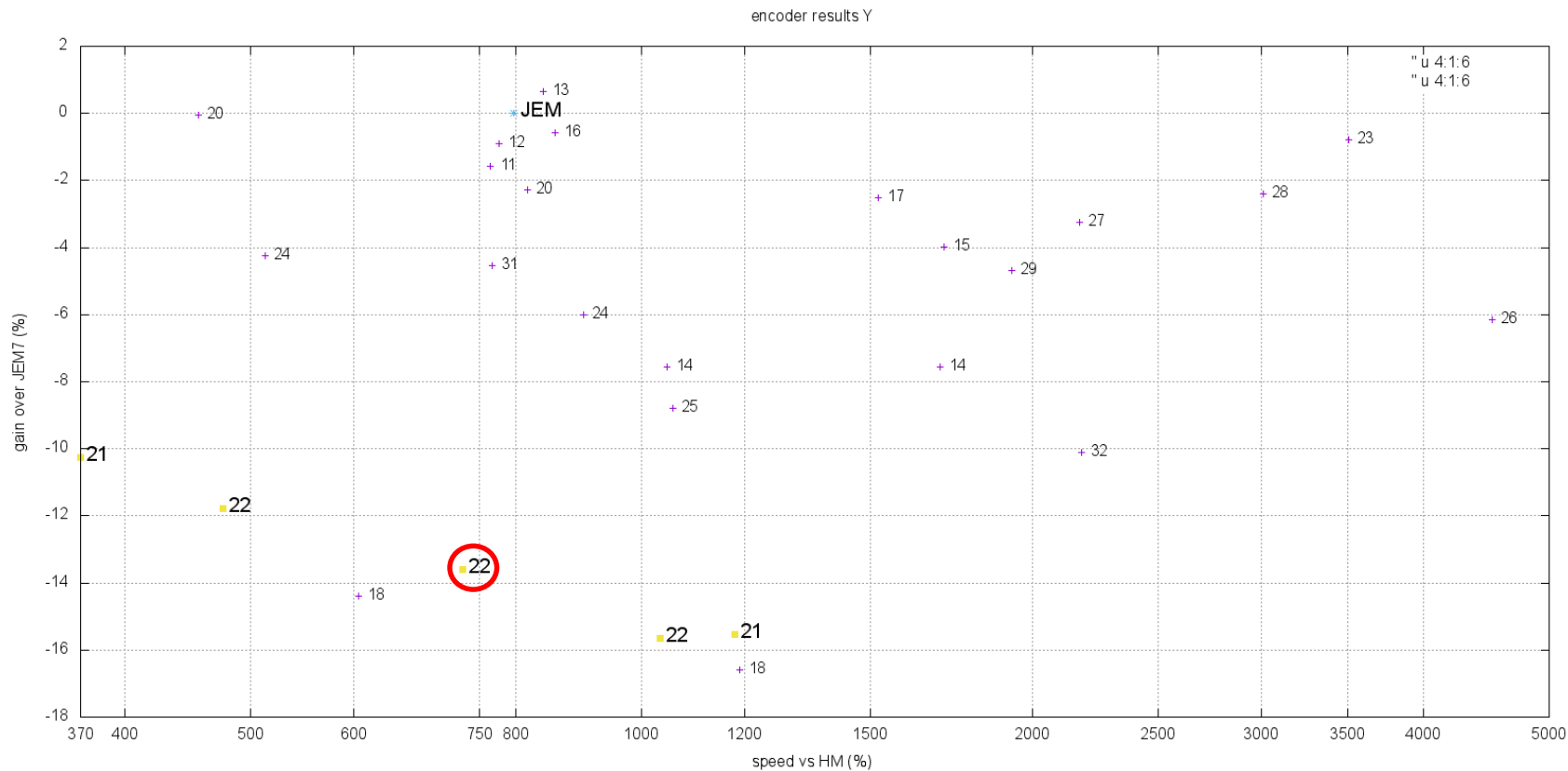
On encoder side, complexity controlled thanks to

- Deep-learning based prediction of split modes in Intra slices, coupled with
- Heuristics and caching mechanisms in RD decisions

Ability to reach various and competitive trade-offs coding gain/complexity

Conclusion

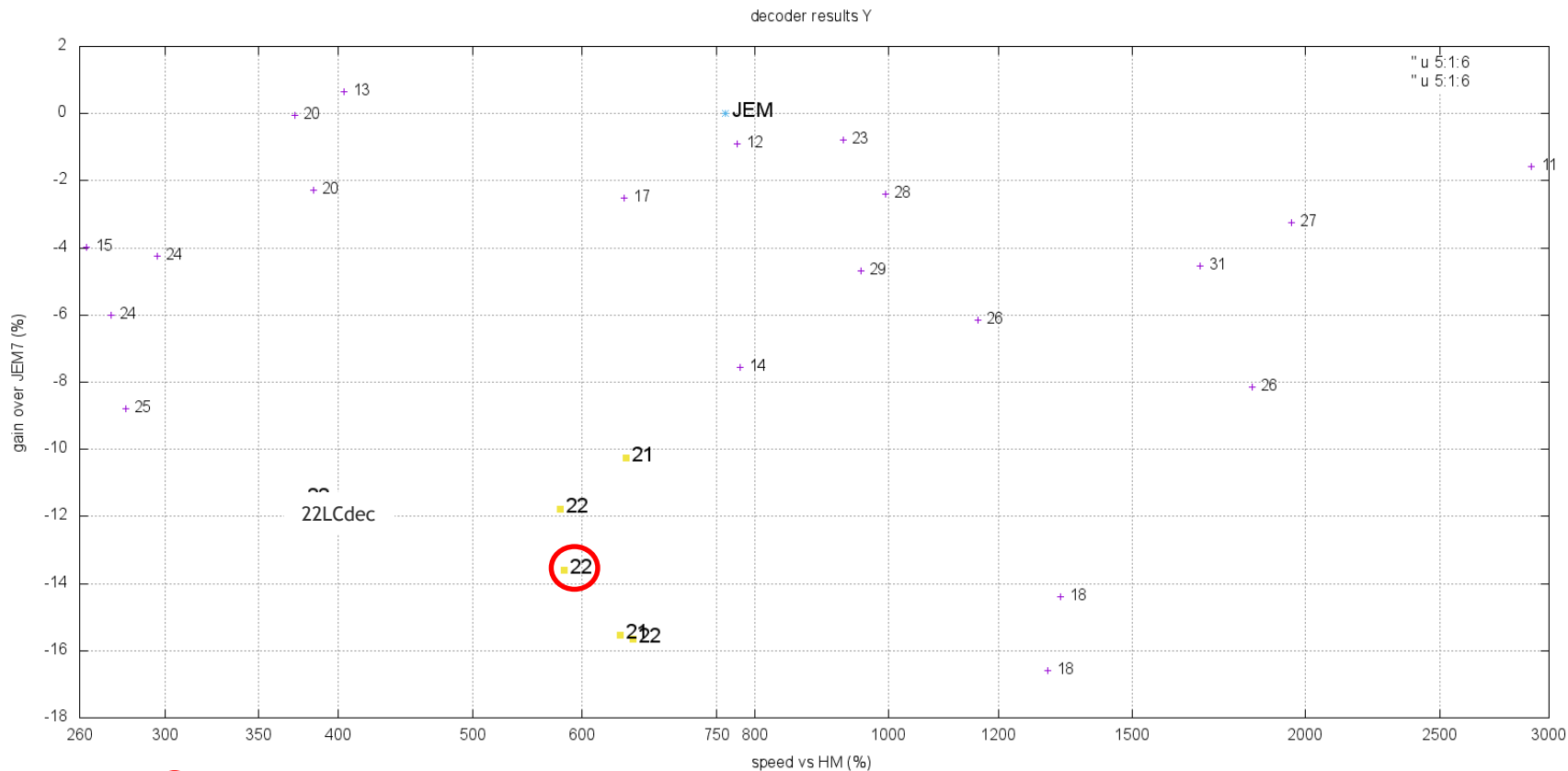
Overview of JVET-J0022 trade-offs - BD-rate-over-JEM7 = f (enc.time over HM)



○ Version used for cfp bitstreams

Conclusion

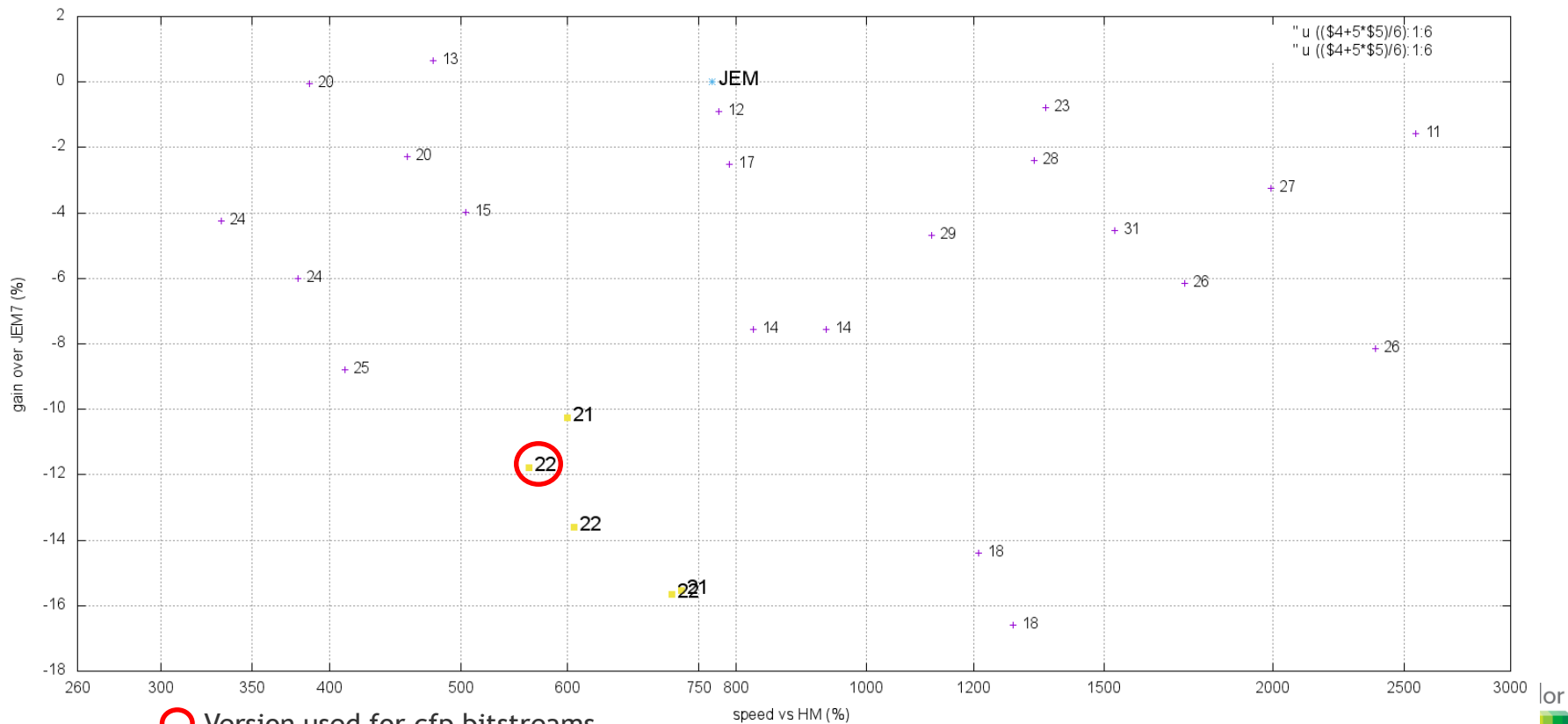
Overview of JVET-J0022 trade-offs - BD-rate-over-JEM7 = f (dec.time over HM)



Conclusion

Overview of JVET-J0022 trade-offs - BD-rate-over-JEM7 = $f(\text{enc.time} + 5 \times \text{dec.time})$

(encoder+5*decoder)/6 results Y



Software corresponding to the CfP response provided

- same basis as the JVET-J0021 MTT version, with the additional tools presented in this contribution
- implementation based on modern C++ programming, and follows the guidelines from JCTVC-H1001
- integrates CNN code that can be used as an inference engine by the encoder, without any dependency to external library

A base software version of MTT also proposed in JVET-J0075

- HM-like + QTBT + ABT + TT
- Could be the starting point for further JVET developments (TEs/CEs)