
JVET-J0014

Description of SDR, HDR, and 360° Video Coding Technology Proposal by Fraunhofer HHI

Overview

- Hybrid video codec (similar to HEVC and JEM)
- Additional coding tools
 - Block partitioning
 - Intra- and inter picture prediction
 - Filtering of prediction signals
 - Transform, quantization and entropy coding
- Includes various JEM tools

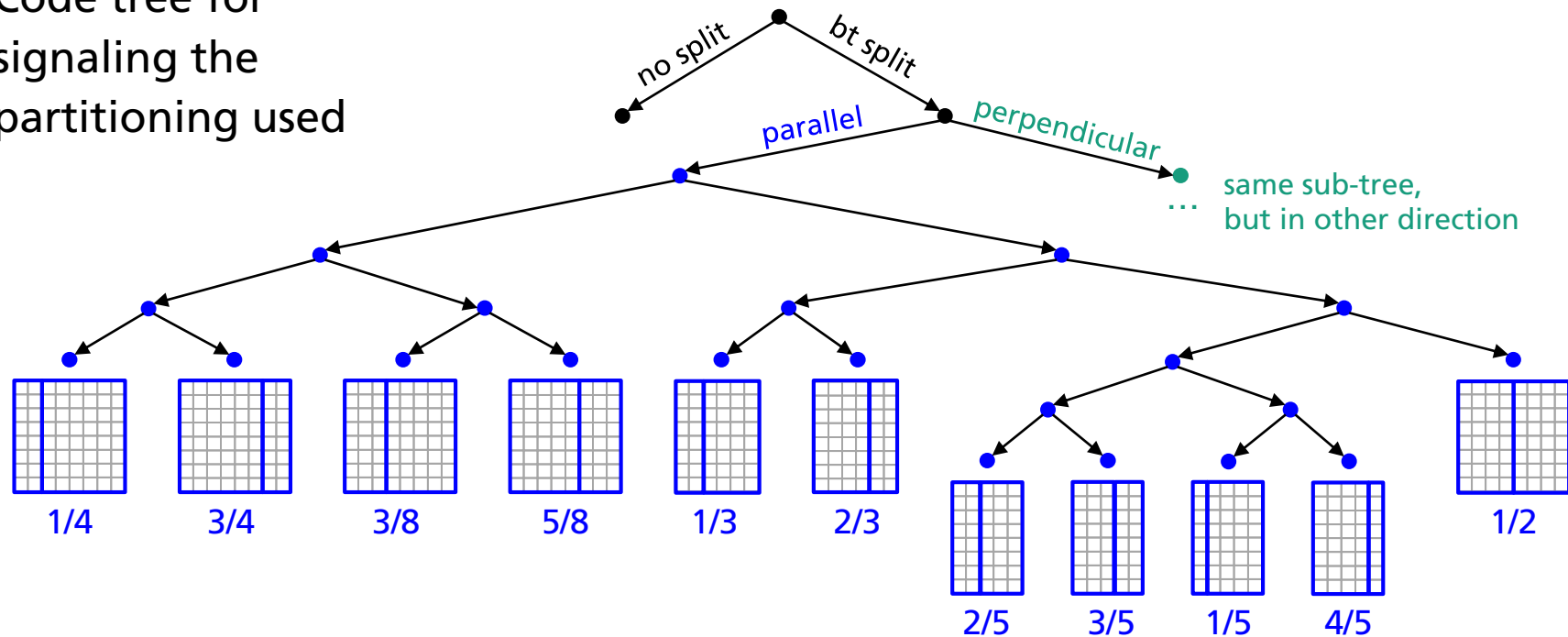
Block Partitioning

Generalized binary block partitioning

- 128×128 CTUs
- Horizontal and vertical splits at different locations (depending on dimension d)
 - $d = 2^N$: 1/2, 1/4, 3/4, 3/8, 5/8
 - $d = 3 \cdot 2^N$: 1/3, 2/3
 - $d = 5 \cdot 2^N$: 1/5, 2/5, 3/5, 4/5
- Width and height of resulting blocks are integer multiples of 4
- Syntax prevents redundant representations of certain partitionings
- Added transform sizes of 12, 20, 24 and 40, 48, 80, 96 (with zeroing of high freq.)
- Used partitioning options can be set using SPS syntax elements

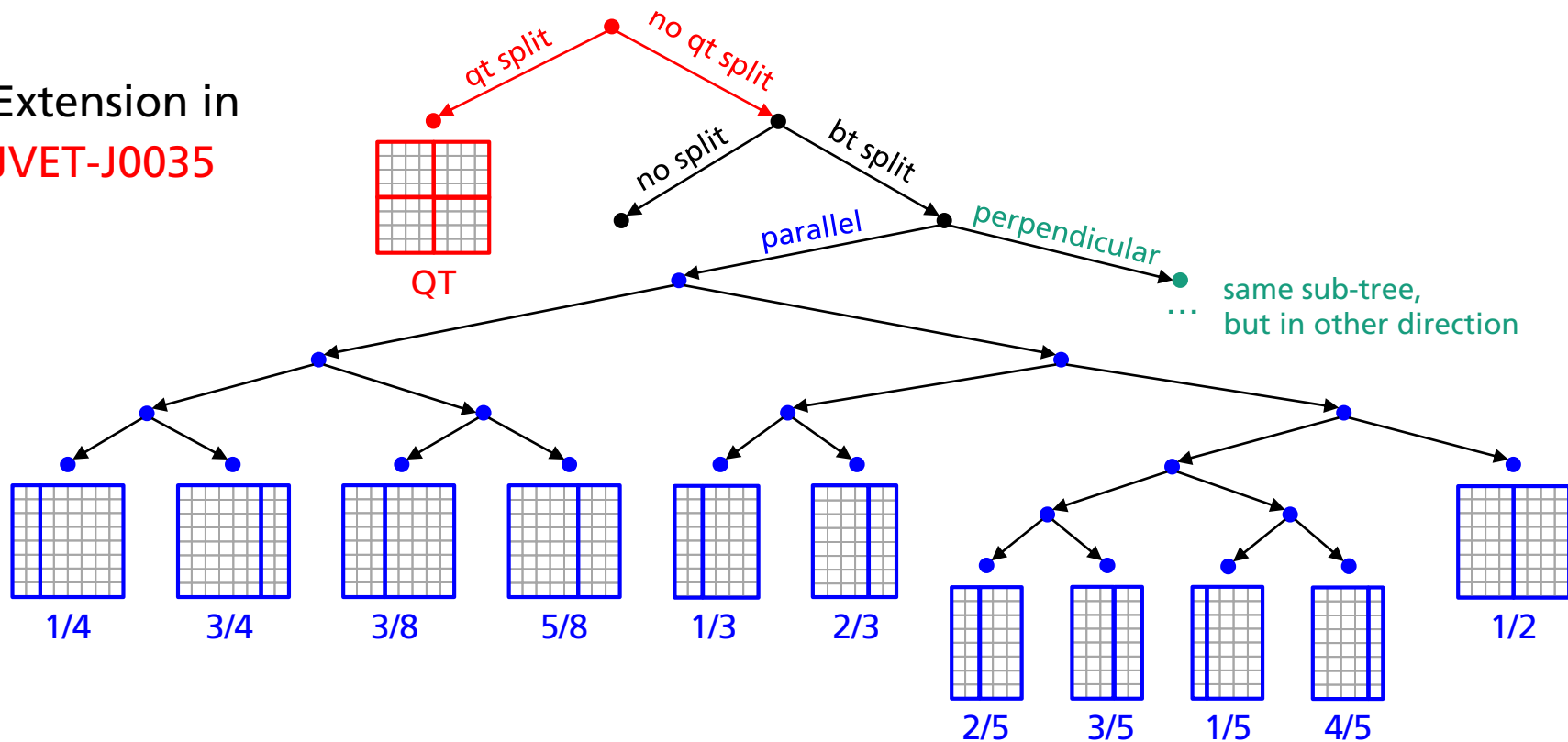
Block Partitioning

Code tree for
signaling the
partitioning used



Block Partitioning

Extension in JVET-J0035



Intra-Picture Prediction and Coding

Proposed Coding Tools

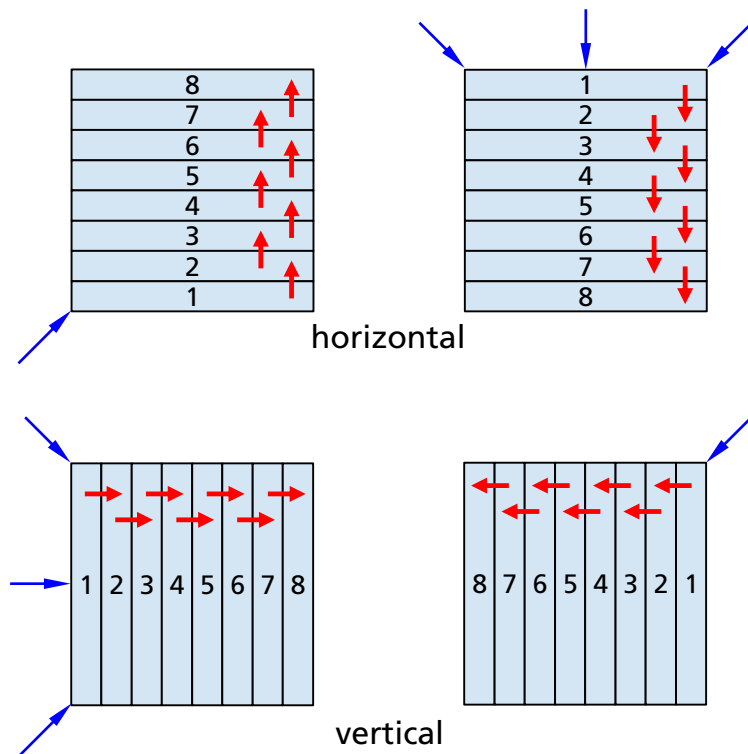
- Line-based intra coding mode
- Intra prediction using neural networks
- Intra region-based template matching
- Bi-lateral filter for reference sample smoothing
- Multi-reference line intra prediction

JEM Coding Tools

- 67 intra prediction modes (65 angular mode)
- 4-tap filter for intra sample prediction
- Intra boundary filtering
- Intra Planar PDPC

Line-based Intra Coding Mode

- Split block into 1d rows or columns
- Subblocks are treated as conventional blocks for intra prediction and transform coding
- Same intra pred. mode for all subblocks
- Luma only (no horizontal or vertical pred.)
- Transform: DCT-II or DST-VII (planar)
- Coding order depends on intra pred. mode
- Signaling: 2 bins
 - Usage of line-based mode
 - Horizontal / vertical split



Intra Prediction using Neural Networks: Sample Prediction

- 35 (or 11) additional luma pred. modes

- 3 hidden layers: $K=2*(N+M+2)$ nodes

- $K \times K$ matrix mult. + K offsets

- ELU look-up table

- Hidden layer are shared by all modes

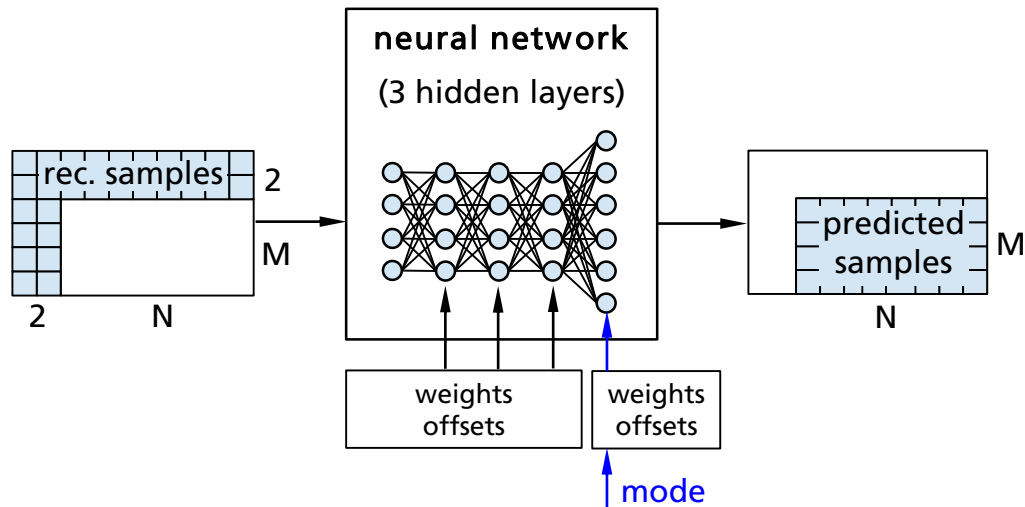
- Block size dep. weights & offsets

- Output layer

- $K \times (NM)$ matrix mult. + offsets

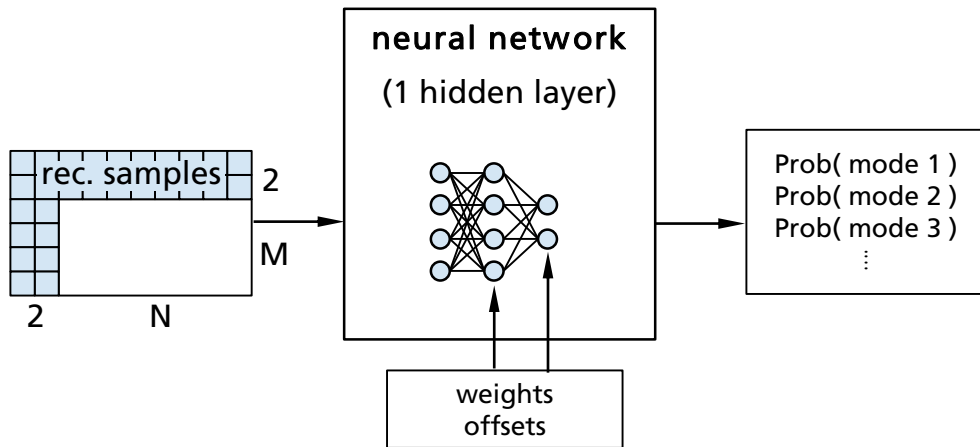
- Block size and mode dependent weights and offsets

- Blocks greater 32×32 are downsampled first (optional for 32×32 blocks)



Intra Prediction using Neural Networks: Mode Coding

- Coded mode id k specifies index into ordered list of modes
 - Ordering depends on neighboring samples
- Hidden layer: $K=2*(N+M+2)$ nodes
 - $K \times K$ matrix mult. + K offsets
 - ELU look-up table
- Output layer (n = number of modes)
 - $K \times n$ matrix mult. + offsets
 - Ordering of n entries



Improvement in JVET-J0037: Prediction in DCT domain

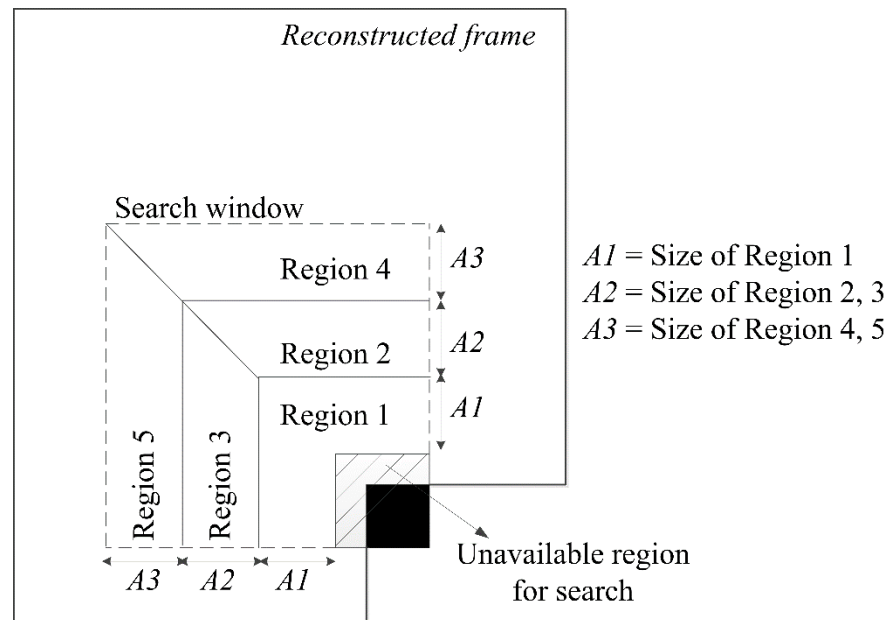
Intra Region-based Template Matching

- Additional coding mode (luma only)
- Search area is split into 5 regions
 - Region index is transmitted
 - Decoder: SSD search in one region
- Derivation of luma prediction signal
 - Find best three matches P_1, P_2, P_3 inside the signaled region
 - Prediction signal given by

$$P_{\text{final}} = (2 \cdot P_1 + P_2 + P_3 + 2) \gg 2$$

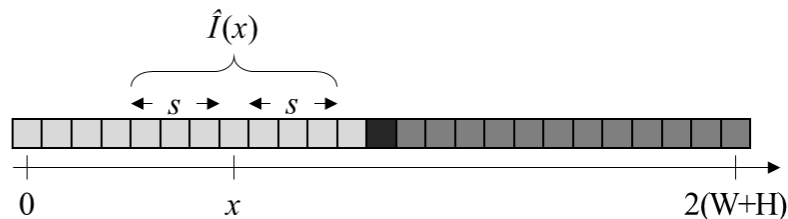
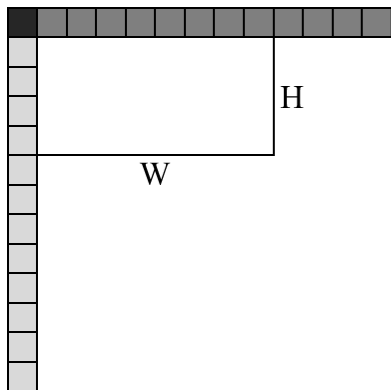


Template: Samples above and to the left of the current block



More details: [JVET-J0039](#)

Bilateral Filter for Intra Reference Sample Smoothing

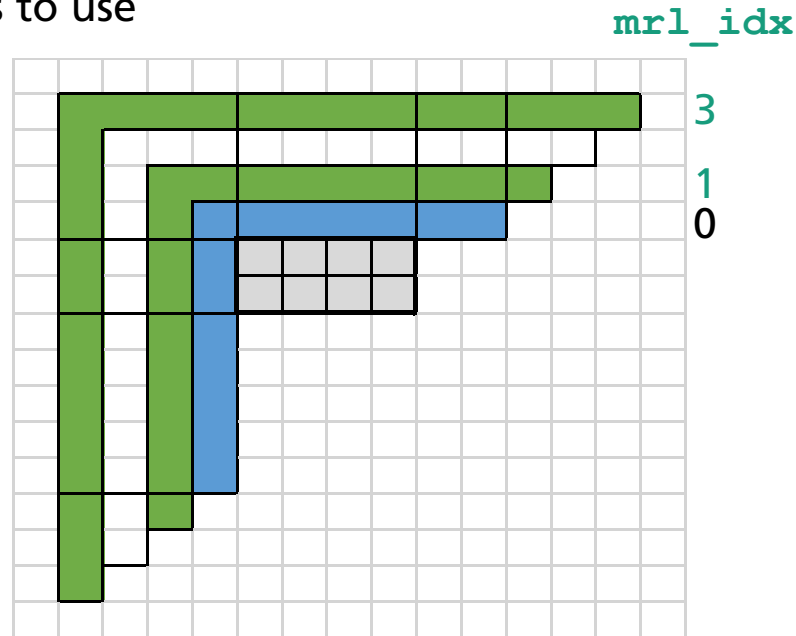


$$\hat{I}(x) = \frac{\sum_{\Delta x_i \leq s} w(x_i) \cdot I(x_i)}{\sum_{\Delta x_i \leq s} w(x_i)} \quad \text{with} \quad w(x_i) = \exp\left(\frac{|I(x_i) - I(x)|^2}{-2\sigma_V^2} + \frac{\Delta x_i^2}{-2\sigma_P^2}\right)$$

- Gaussian bilateral filter for line of reference samples
- Replaces strong intra smoothing (HEVC / JEM) for large blocks
- Goal: Reduce noise while preserving edges
- Look-up table for filter weights (no complex calculations of exponential function)

Multi-Reference Line Intra Prediction

- Use 2 additional reference lines
- `mrl_idx` indicates which reference samples to use
- Only applied for luma
- Reference samples used for:
 - DC
 - PLANAR
 - DIR
- If `mrl_idx > 0`
 - No planar PDPC
 - No boundary filter
 - No edge filter



Motion-Compensated Prediction

Proposed Coding Tools

- Multi-Hypothesis Prediction
- Restricted Merge Mode
- Multi-directional Boundary Padding

JEM Coding Tools

- Sub-PU level motion vector derivation
- Locally adaptive motion vector resolution
- 1/16-th luma sample accurate motion vectors
- Overlapped block motion compensation
- Local illumination compensation
- Affine motion-compensated prediction
- Pattern matched motion vector derivation
- Decoder-side motion vector refinement
- Bi-directional optical flow

Multi-Hypothesis Prediction

- Superposition of more than 2 motion-compensated prediction signals with different weights for individual prediction signals (“hypotheses”)
- Submitted bitstreams: Restricted to one additional hypothesis
- Recursive approach (extendable to more than one additional hypothesis)

$$p_{\text{final}} = (1 - \alpha) \cdot p_{\text{uni|bi}} + \alpha \cdot p_{\text{add}}$$

- Combined reference picture list for additional hypotheses p_{add}
- Not supported in SKIP mode (but in MERGE mode)
- Two weights supported: $\alpha \in \left\{\frac{1}{4}, -\frac{1}{8}\right\}$

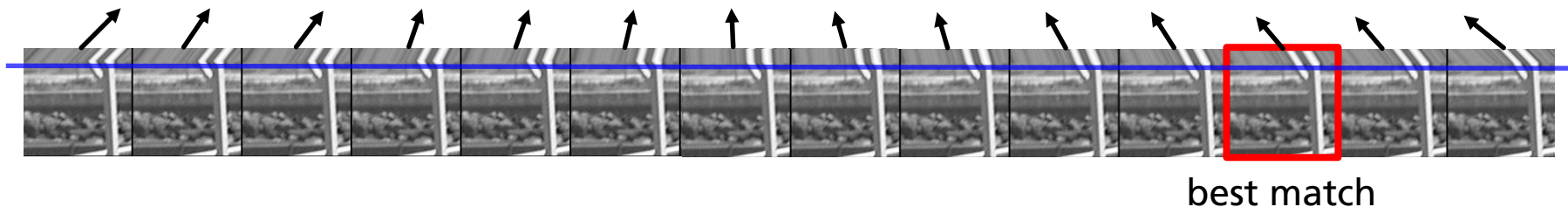
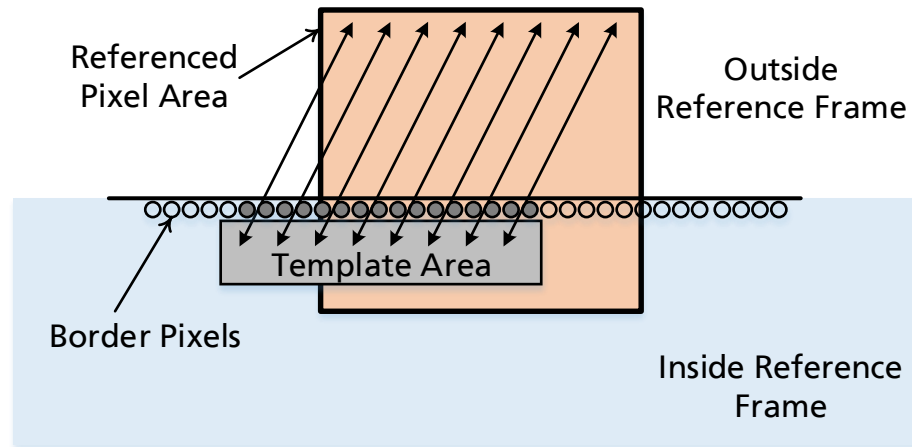
More details: JVET-J0041

Restricted Merge Mode

- Idea: Merge mode with restriction to uni-prediction
- Merge mode (not SKIP mode) in B slices
 - Additional syntax element `merge_ref_pic_list_idc`
 - Possible values: NONE, LIST_0, LIST_1
- When `merge_ref_pic_list_idc != NONE`
 - Remove candidates with uni-prediction
 - Modify prediction utilization flags for bi-prediction candidates
 - Fill list with zero motion vectors (iterating over reference picture list)

Multi-Directional Boundary Padding

- Find better continuation of reference frame border for motion compensation
- Use angular intra prediction (half of the angular modes)
- Estimate best angular mode:
 - Decoder: Based on template area inside reference frame (using SAD)
 - Encoder: After conventional motion search



Prediction Signal Filtering

Proposed Coding Tools

- Diffusion filter
- DCT thresholding

Note:

- Filters are applicable to intra- and inter-picture prediction
- Submitted bitstreams: Filters only used for intra (in restricted setting)

Diffusion Filter

- Iterative filters for luma prediction signal

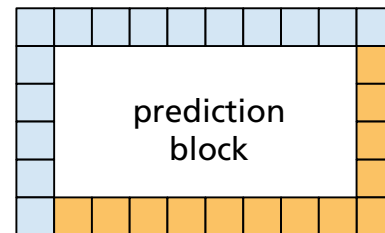
- **Linear filter mode:** In each iteration

$$p_{n+1} = p_n * \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- **Non-Linear filter mode:**

- Goal: Less smoothing across edges
- Use position depend 3×3 filter kernels $h(x,y)$ in the iterations
- Filter kernels $h(x,y)$ are derived based on sample differences (in x and y direction)
- Same filters kernels are used in all iterations

rec. border samples

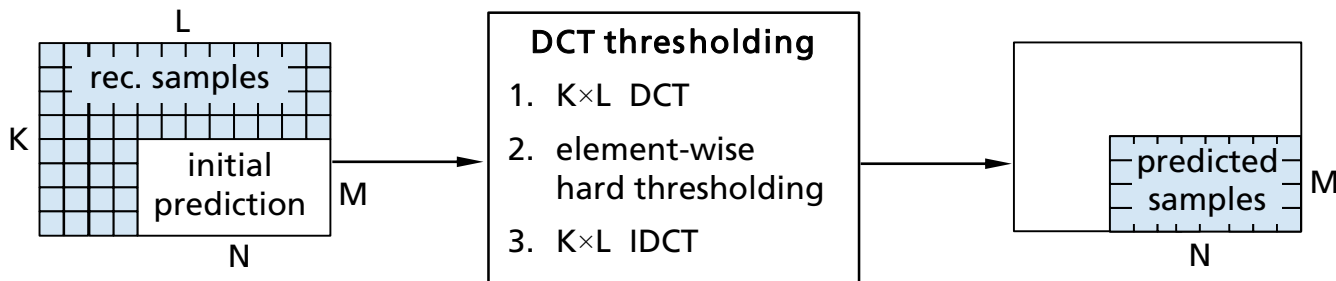


reflected samples

filter idx	filter type	iterations	
		intra	inter
0	no filter	n/a	n/a
1	linear	5	5
2	linear	15	35
3	non-linear	5	4
4	non-linear	20	8

More details: JVET-J0038

DCT Thresholding



- Syntax includes parameters for extension size and threshold
- Re-use transform sizes available for generalized binary splits
- Fixed threshold t for all transform coefficients

$$y = \begin{cases} 0 & : |x| \leq t \\ x & : |x| > t \end{cases}$$

Transforms, Quantization, and Entropy Coding

Proposed Coding Tools

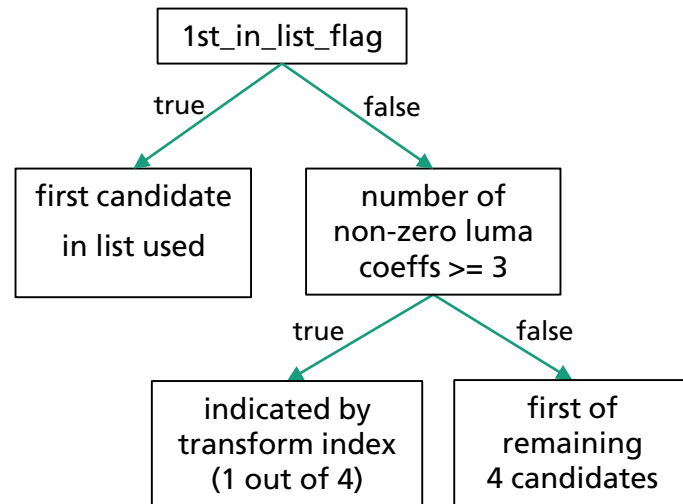
- Improvement of adaptive transforms for intra blocks
- Dependent scalar quantization
- Transform coefficient coding (incl. modifications for dependent quantization)
- Multi-parameter coding engine with context-specific window sizes
- CTU-line based context initialization from previous frames
- Modified QP prediction

JEM Coding Tools

- Adaptive multi-core transforms for inter blocks

Adaptive Transforms for Intra-Predicted Blocks

- 5 transform candidates for each block (EMT, NSST, add. non-separable transforms)
- List of transform candidates depends on
 - prediction mode, channel type, and block size
- 4×4, 4×8, 8×4, and 8×8 blocks
 - primary transform only (can be non-separable)
- Blocks with width or height greater 8
 - separable primary transform
 - non-separable secondary transform (4×8, 8×4, or 8×8)
- Signaling: Two bins

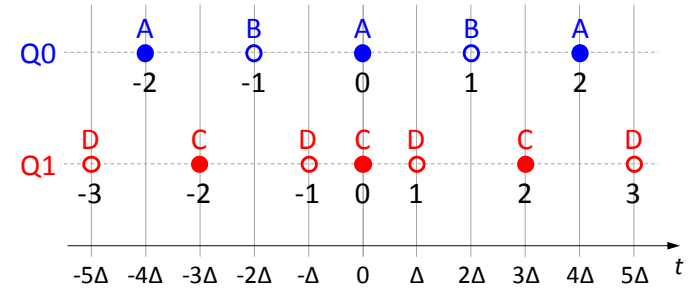


Dependent Quantization

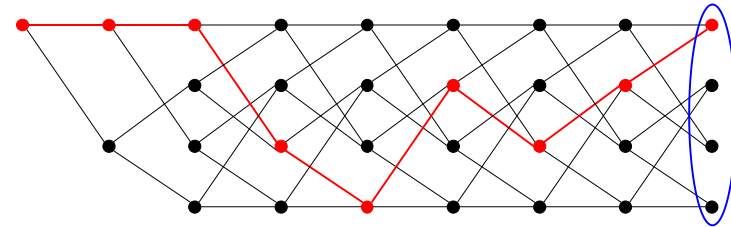
- Two scalar quantizers (Q0, Q1) for all coefficients
- Quantizer used: determined by state machine
 - 4 states (unique mapping to quantizer)
 - State transition determined by parity of previous level k
- Decoder: Additional mapping (in coding order)

```
for( state = 0; scanIdx = last; scanIdx >= 0; scanIdx-- ) {
    k = transCoeffLevel[ scanIdx ];
    qIdx[ scanIdx ] = ( k << 1 ) - ( state > 1 ? sign(k) : 0 );
    state = stateTransTable[ state ][ k & 1 ];
}
```

- Encoder: Pre-quantization (2 cand. per quant.) + Viterbi algorithm for 4 state trellis

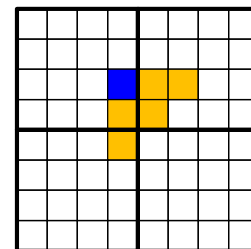
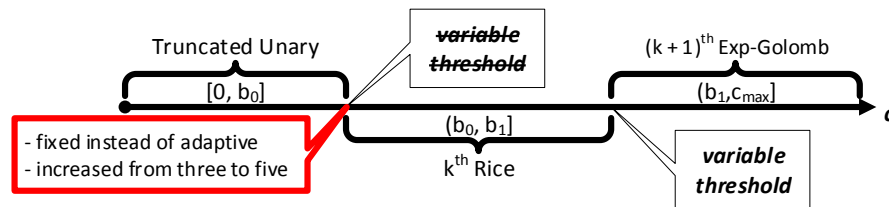


state	quant.	next state for ...	
		$(k \& 1) == 0$	$(k \& 1) == 1$
0	Q0	0	2
1	Q0	2	0
2	Q1	1	3
3	Q1	3	1



Coding of Transform Coefficient Levels

- Coding order
 - Single pass over absolute transform coefficients
- Binarization
 - Fixed first binarization bound (=5)
 - Add. greater 3 and greater 4 flags
- Context modelling: Context depends on
 - Diagonal / whether last coefficient
 - Sum of absolute levels in local template (sig flag)
 - Sum of absolute values greater 1 (gtX flags)
 - also used for Rice parameter
 - Quantizer (Q0 or Q1) used (for sig flag, gt1 flag)



Arithmetic Coding Engine

- Counter-based probability estimation with custom window sizes

$$c_{t+1} = \begin{cases} c_t + ((2^{15} - c_t) \gg n), & \text{if } x_t = 1 \\ c_t - (c_t \gg n) & , \text{if } x_t = 0 \end{cases}$$

- Two 15-bit counters c (as in JEM), but with custom window sizes $n \in [1; 11]$
- Window sizes depend on slice type and cabac init flag (not modified during coding)

- Adjustment of coding engine

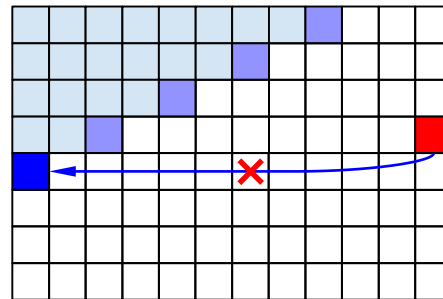
- 32×8 range LPS table with 8-bit values (instead of 512×64 table with 9-bit values)
- LPS table indexed by $\text{stateIdx} = ((c - 2^{14}) \wedge (\text{mps} - 1)) \gg 9$ with $\text{mps} = c \gg 14$

- Context initialization from previous frame (as in JEM)

- One context set per CTU line

Modified QP Prediction

- QP prediction in HEVC / JEM
 - Uses QP of previous CTU in coding order in some situations
 - Can be a problem in encoder-only WPP (prevents in-place deblocking)



- Modified QP prediction

$$p = \text{Clip3}(\min(a, b), \max(a, b), a + b - c)$$

- Use slice QP for non-available values

$c(\text{orner})$	$a(\text{bove})$
$b(\text{efore})$	$p(\text{redicted})$

In-Loop Filtering

Proposed Coding Tools

- Improvement of adaptive loop filter

JEM Coding Tools

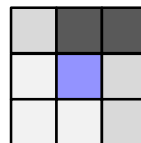
- Bilateral in-loop filter
- Content-adaptive clipping

Multiple-Feature based Adaptive Loop Filter

- Similar to GALF, but 3 options for classification

1. GALF classification (as in JEM): 25 classes
2. Sample value based classification: 27 classes (27 bands of dynamic range)
3. Rank based classification: 27 classes (3 bands × 9 ranks)

- Rank: Sample value is compared to its 8 direct neighbors



$$r = \sum_k (s' < s'_k ? 1 : 0)$$

- Filtering and Signalization

- Filtering and transmission of filter coefficients is identical to GALF (JEM)
- Additional: Slice header syntax element specifying classification used
- Encoder: Test all three classifications

Encoder Control

Encoder Decision Algorithm

- Same Lagrangian encoder control as in HM / JEM
- Some strategies for encoder speed-up (similar level as in JEM)

QP Assignment

- First submission
 - Fixed QP assignment with QP cascading (same as in HM/JEM anchors)
 - HDR: QP as function on average luma level (same as in HM/JEM anchors)
- Second submission
 - Perceptually optimized QP adaptation
 - HDR: Includes QP adaptation based on average luma level

Only difference between the two submissions: QP assignment

Perceptually Optimized QP Adaptation

- Basic approach: Similar to approach presented in JVET-H0047
 - Derive activity measure per CTU

$$a_k = \max \left(1; \left(\frac{1}{|B_k^*|} \sum_{(x,y) \in B_k^*} |h(x,y)| \right)^2 \right)$$

with $h(x,y) = 4 \cdot s(x,y) - s(x-1,y) - s(x+1,y) - s(x,y-1) - s(x,y+1)$

- Set CTU QP and lambda according to

$$QP_k = QP_{\text{slice}} + \text{round} \left(2 \cdot \log_2 \left(\frac{a_k}{a_{\text{ref}}} \right) \right) \quad \text{and} \quad \lambda_k = \lambda_{\text{slice}} \cdot 2^{\frac{QP_k - QP_{\text{slice}}}{3}}$$

- For HD and base QP < 42: Reduce CTU size to 64x64

HDR

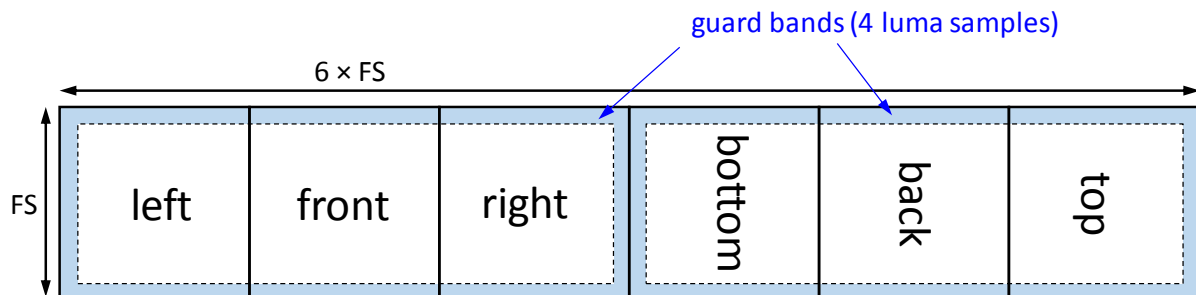
- No HDR specific coding tools
- No HDR pre- or postprocessing

360° Video

- No 360° video specific coding tools
- No pre- or postprocessing (except projection)

Projection format

- Equi-angular cubemap (EAC)
- Guard bands with a width of 4 luma samples (as suggested in JVET-G0071 for ACP)
- Back projection: Blending of samples that are available in more than one face



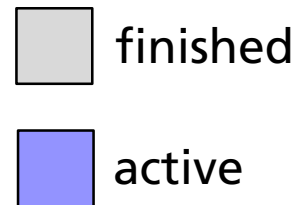
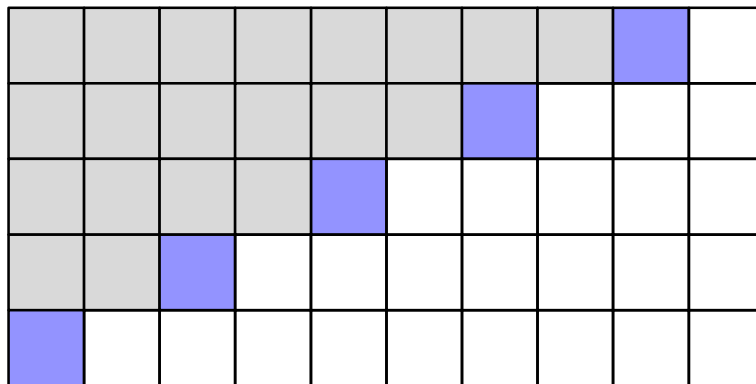
original ERP format	face size	coded frame size
6144 × 3072	1152 × 1152	6912 × 1152
8192 × 4096	1280 × 1280	7680 × 1280

Software

- Basis: NextSoftware presented in JVET-H0047 and JVET-I0034
 - Alternative implementation of JEM tools
 - Improved code readability
 - Run time configurability
 - Improved debugging options
- Proposal: Extension of NextSoftware
 - Added partitioning options (triple split, gen. binary split, QT+BTS)
 - Integration of additional coding tools
 - Thread parallel encoding

Thread Parallel Encoding

- Option 1: Encoder-only wavefront parallel processing



- Option 2: Parallel testing of split modes
- Both options can be combined

More details: [JVET-J0036](#)

Results of Submission 1: SDR

constraint set 1

class	rel. to HM anchors			rel. to JEM anchors		
	BDR-Y	BDR-U	BDR-V	BDR-Y	BDR-U	BDR-V
SDR-A	-39.6%	-45.8%	-45.1%	-6.3%	-5.9%	-5.5%
SDR-B	-36.5%	-47.9%	-48.0%	-8.8%	-7.9%	-6.4%
average	-38.1%	-46.9%	-46.5%	-7.5%	-6.9%	-6.0%

constraint set 2

class	rel. to HM anchors			rel. to JEM anchors		
	BDR-Y	BDR-U	BDR-V	BDR-Y	BDR-U	BDR-V
SDR-A	-29.7%	-46.5%	-45.5%	-7.2%	-7.6%	-5.7%

Results of Submission 1: HDR and 360° Video

HDR (BD rates based on PSNR)

class	rel. to HM anchors			rel. to JEM anchors		
	BDR-Y	BDR-U	BDR-V	BDR-Y	BDR-U	BDR-V
HDR-A	−32.5%	−64.0%	−67.3%	−6.2%	−7.1%	−3.7%
HDR-B	−32.8%	−61.3%	−52.6%	−9.0%	−23.4%	−18.1%
average	−32.7%	−62.3%	−58.1%	−8.0%	−17.3%	−12.7%

360° video (BD rates based on E2E WS-PSNR)

class	rel. to HM anchors			rel. to JEM anchors		
	BDR-Y	BDR-U	BDR-V	BDR-Y	BDR-U	BDR-V
360°	−35.7%	−52.7%	−53.7%	−15.7%	−16.3%	−14.5%

Comments on Relationship between Luma and Chroma QP

- Some proposals include a change of the relationship between luma and chroma QP
 - Same or similar concept can be used in JEM or other proposals
 - Effect on BD rate savings not clear

- Experiment for CfP conditions

- New bitstreams with modified relationship between luma and chroma QP
- Used mapping table suggested in JVET-J0018 (MediaTek)

luma QP_Y	$QP_Y < 30$	$30 \leq QP_Y \leq 34$	$QP_Y > 34$
chroma QP_C	QP_Y	$QP_Y - 1$	$QP_Y - 2$

- UHD sequences: Additional chroma QP offset = 1 (as in JEM)
- HD sequences: Additional chroma QP offset = 2
- Used updated version of our software (QT+BTS partitioning + other improvements)

Effect of Chroma QP change: SDR Constraint Set 1

rel. to HM anchors

class	submission 1 (Y, U, V)			with chroma QP change (Y, U, V)			Δ BDR-Y
SDR-A	-39.6%	-45.8%	-45.1%	-46.1%	-30.4%	-31.6%	-6.5%
SDR-B	-36.5%	-47.9%	-48.0%	-38.5%	-20.7%	-20.9%	-2.0%
average	-38.1%	-46.9%	-46.5%	-42.3%	-25.6%	-26.3%	-4.2%

rel. to JEM anchors

class	submission 1 (Y, U, V)			with chroma QP change (Y, U, V)			Δ BDR-Y
SDR-A	-6.3%	-5.9%	-5.5%	-15.0%	+15.0%	+11.7%	-8.7%
SDR-B	-8.8%	-7.9%	-6.4%	-11.0%	+31.8%	+33.5%	-2.2%
average	-7.5%	-6.9%	-6.0%	-13.0%	+23.4%	+22.6%	-5.4%

Summary

- Proposed Coding Tools for
 - Block partitioning (generalized binary splits)
 - Intra-picture prediction (neural networks, template matching, ...)
 - Inter-picture prediction (multi-hypotheses pred., restricted merge, boundary ext.)
 - Transform coding (adapt. transforms, dependent quantization)
 - Entropy coding (transform coefficients, coding engine with custom window sizes)

- Objective results
 - SDR CS1: -38.1% (vs HM), -7.5% (vs JEM) [mod. CQP: -42.3% (HM), -13.0% (JEM)]
 - SDR CS2: -29.7% (vs HM), -7.2% (vs JEM)
 - HDR: -32.7% (vs HM), -8.0% (vs JEM)
 - 360°: -35.7% (vs HM), -15.7% (vs JEM)