



JVET-Q0199 AHG8: Support of ROI (Region-of-Interest) RPR

17th JVET Meeting: Brussels, BE, 7–17 January 2020

Dolby Laboratories, Inc.

Outline

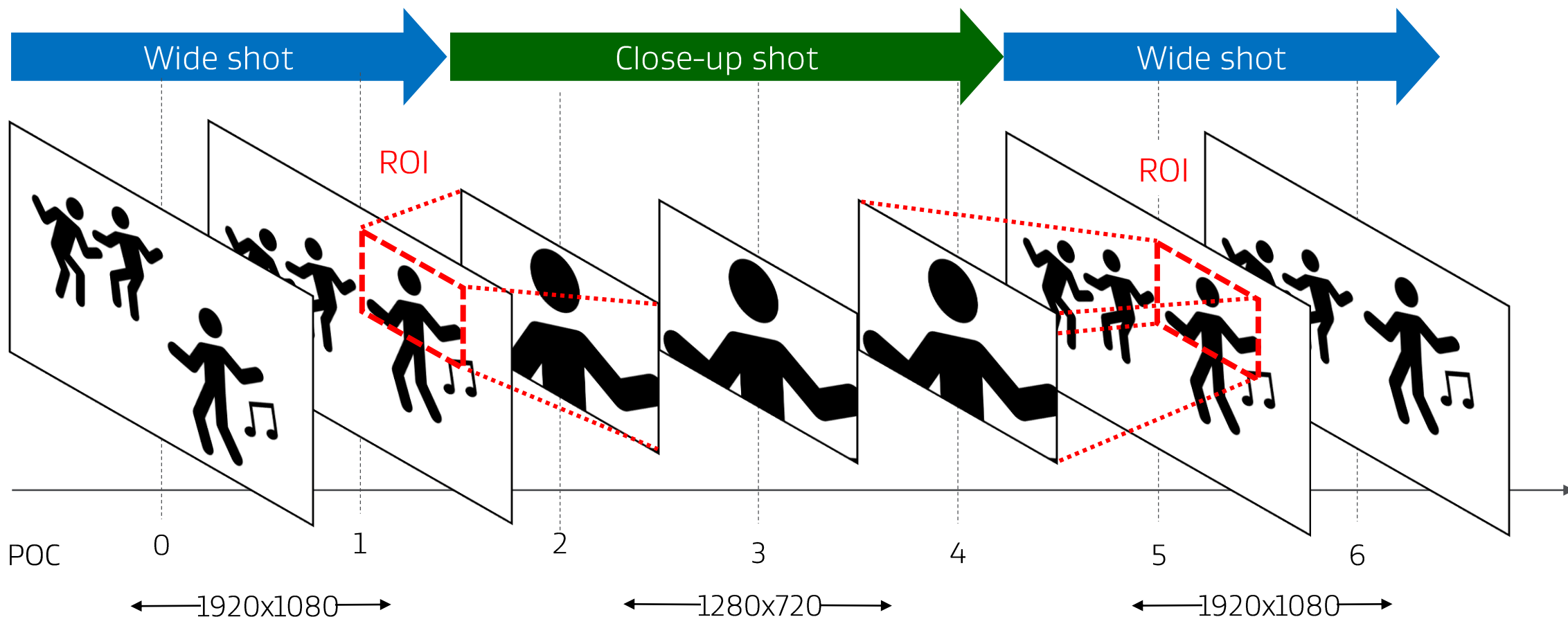
- Conceptual overview of region-of-interest (ROI) RPR
- ROI RPR design and constraints
- ROI RPR syntax, semantics and changes to decoding processes
- ROI RPR software implementation and simulations
- Memory bandwidth constraints
- Conclusion



Conceptual overview of ROI RPR

Example: Region-of-interest reference picture resampling (ROI RPR)

Support for resampling of subregions within the RPR framework



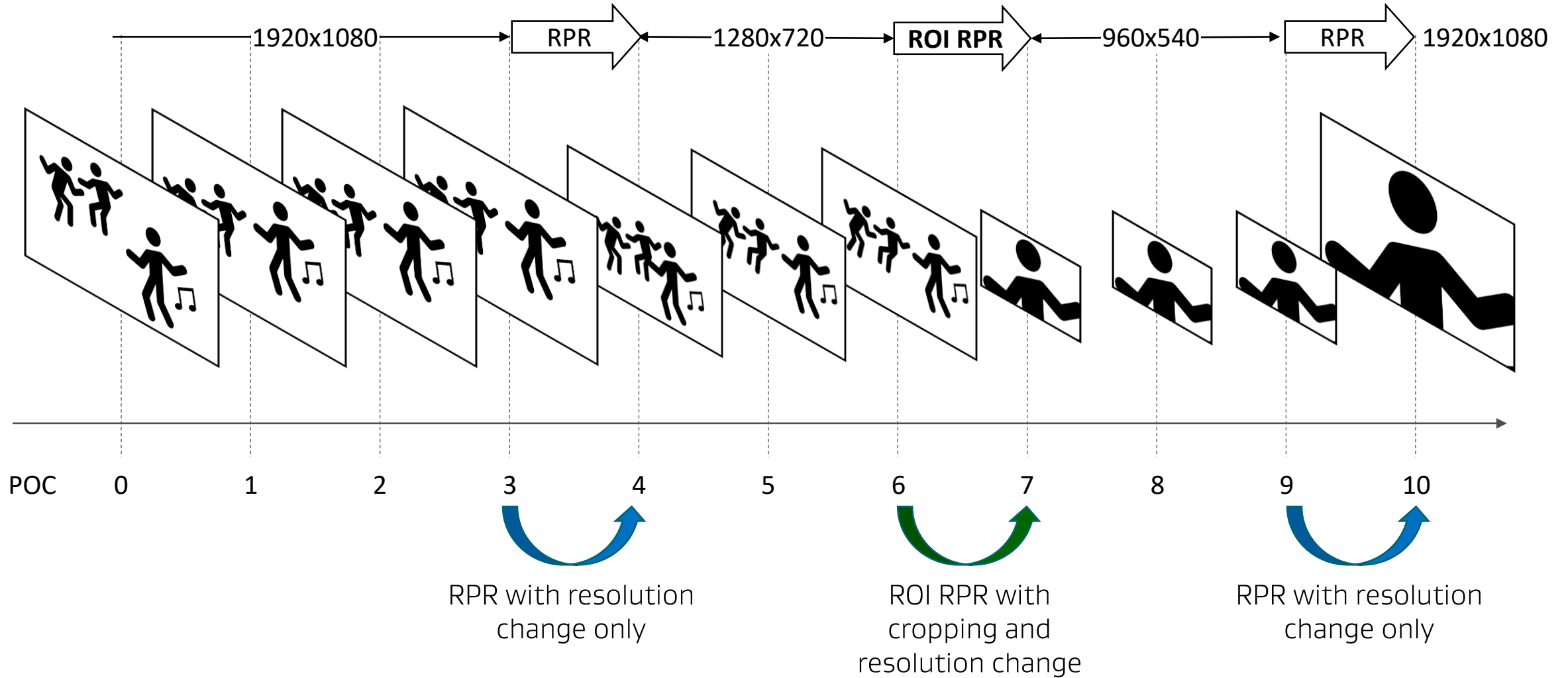
Support for different picture resolution

Key concepts for ROI RPR

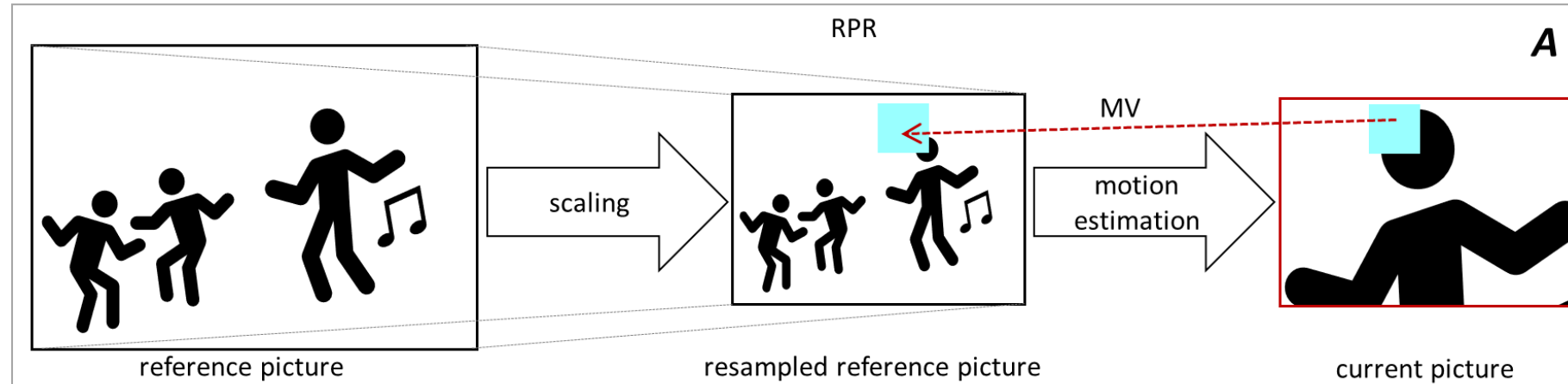
- Proposes syntax and semantics to support RPR for subregions (ROI) within a picture.
 - RPR currently supports only the entire picture, not subregions.
 - Decoding process change is minor.
- ROI RPR leverages RPR as currently specified in VVC.
 - ROI RPR may be thought of as a generalization of RPR from decoder perspective.
- ROI RPR addresses meeting notes for JVET-P0336 on support of ROI scalability using RPR.
 - Concern was expressed about whether this functionality would have a complexity problem.
 - There is interest in this in concept, further study with provision of text and software was encouraged.
- Test results indicate average coding improvement of ~11% in LDB for video content having a mix of camera close-ups and wide shots.
- ROI RPR supports spatial scalability without additional syntax or semantics.

Example: RPR and ROI RPR in single layer low-delay P

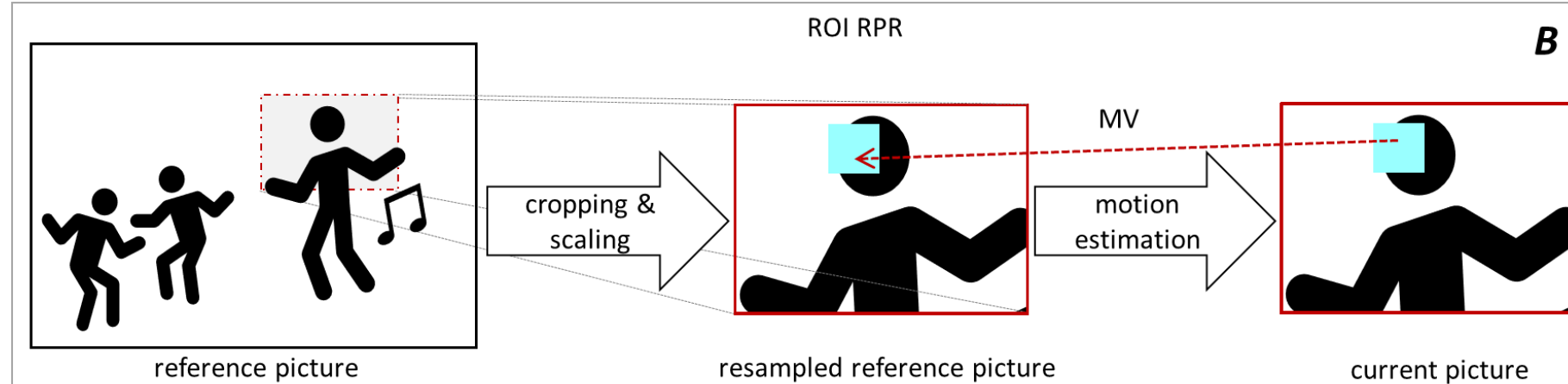
Example: Low-delay P, each frame uses previous frame for inter prediction



Support for ROI RPR (encoder side)



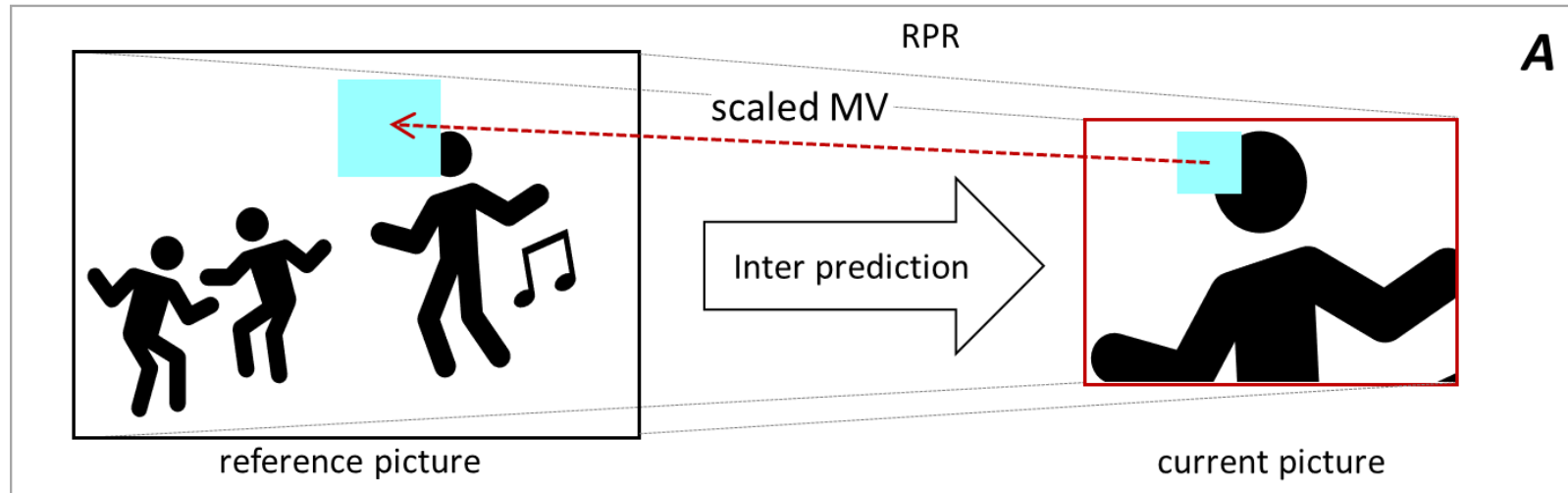
RPR with
resolution change
only
(downsampling
only)



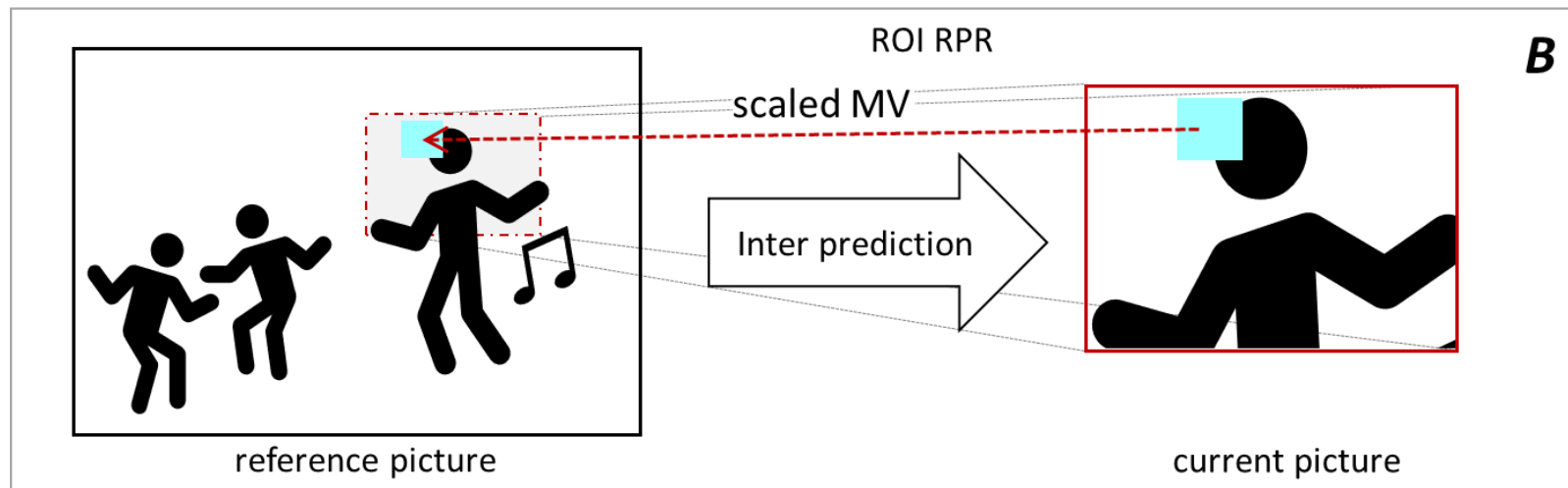
ROI RPR with
cropping and
resolution change

(picture
downsizing with
ROI upsampling)

Support for ROI RPR (decoder side)



RPR with
resolution change
only
(downsampling
only)



ROI RPR with
cropping and
resolution change
(picture
downsizing with
ROI upsampling)



ROI RPR design and constraints

Proposal: ROI Window

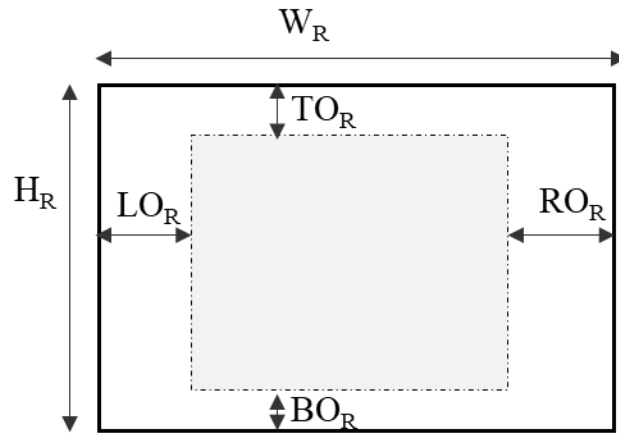
RPR as currently specified

- JVT-P0590 (adopted in VVC draft 7): scaling window offset parameters in PPS
 - Derive scaling ratio of ref pic and cur pic
 - Locate inter prediction correspondence of ref pic and cur pic (fractional sample interpolation process)
-

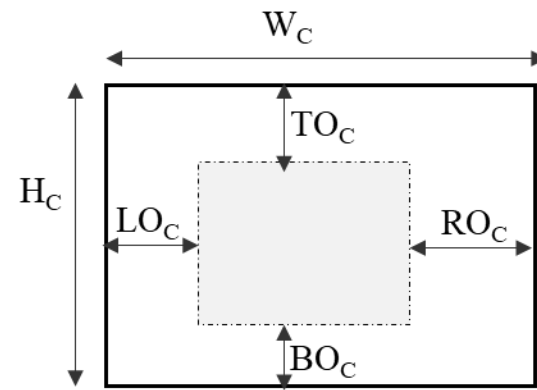
Proposed ROI RPR

- Key concept: replace scaling window offset parameters with ROI window offset parameters when ROI RPR is used.
- New ROI window offset parameters syntax are needed to support ROI RPR
 - ROI window cannot be supported directly by scaling window offset parameters in PPS
 - Scaling window offset parameters are a property of the picture itself (enable pic size of multiple of 8).
 - ROI window offset parameters are specified with respect to both cur and ref pics.

ROI RPR Design



Reference picture (R)



Current picture (C)

- Specify ROI window offset parameters for current and reference pictures
 - Ref pic: (LO_R , RO_R , TO_R , BO_R)
 - Cur pic: (LO_C , RO_C , TO_C , BO_C)
- Determine horizontal and vertical scaling ratios:
 - $hor_scale = (W_R - LO_R - RO_R) / (W_C - LO_C - RO_C)$
 - $ver_scale = (H_R - TO_R - BO_R) / (H_C - TO_C - BO_C)$

ROI RPR Design

Scaling window is used as in default RPR case for a pair of cur pic and ref pic.

- If ROI RPR is enabled, ROI window offset parameters are used in derivation of scaling ratio and in offset handling in fractional interpolation process;
 - Otherwise (ROI RPR is disabled), scaling window offset parameters are applied as in current VVC design.
-

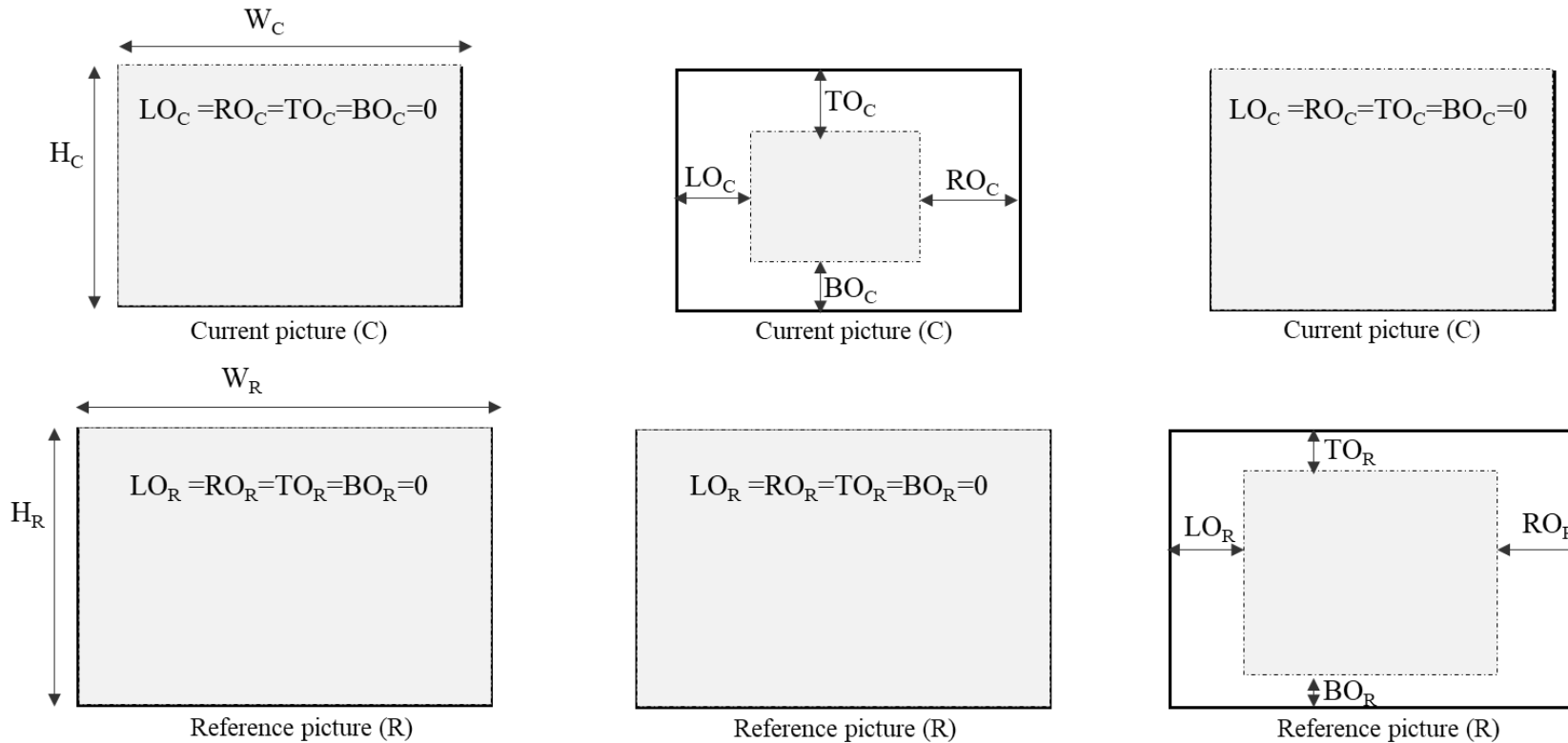
Constraints proposed to simplify design and implementation:

- 1) Any ROI picture is completely contained in the full picture: eliminates the need to check for samples outside the ROI window
- 2) Any picture, current or reference, may contain at most one ROI window: avoid complexity of determining which of several ROI windows should be used for each sample

Note: Constraint 1 is intended to simplify encoder implementation. It would not affect the decoding process. If constraint 1 is not adopted, extra care would be required at encoder to properly handle motion estimation for samples outside of ROI window.

ROI RPR with Constraints 1 and 2

Constraints: 1) Any ROI picture is completely contained in the full picture
2) Any picture, current or reference, may contain at most one ROI window



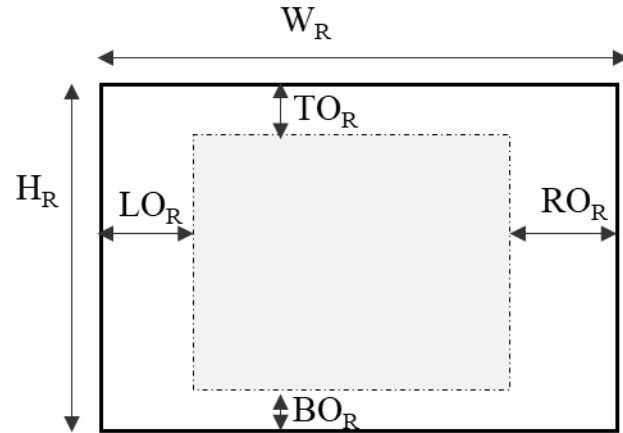
Case 1: ROI RPR same as default RPR mode

Case 2: Entire ref pic corresponds to ROI window in cur pic

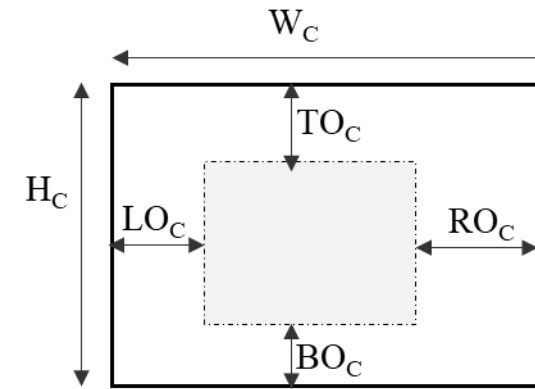
Case 3: ROI window in ref pic corresponds to entire cur pic

ROI RPR with Constraints 2 Only

Constraints: 1) Any ROI picture is completely contained in the full picture
2) Any picture, current or reference, may contain at most one ROI window



Reference picture (R)

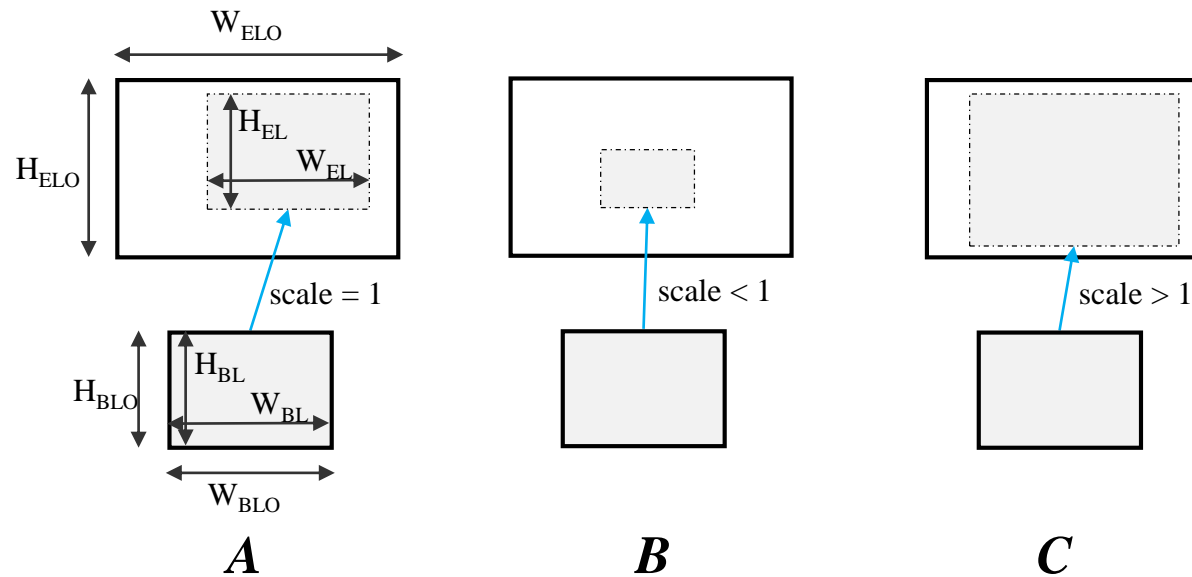


Current picture (C)

- Decoder: same process for all samples as in ROI RPR with both constraint 1 and 2
- Encoder: extra care for ME in VTM7.0 software

ROI RPR and Spatial Scalability

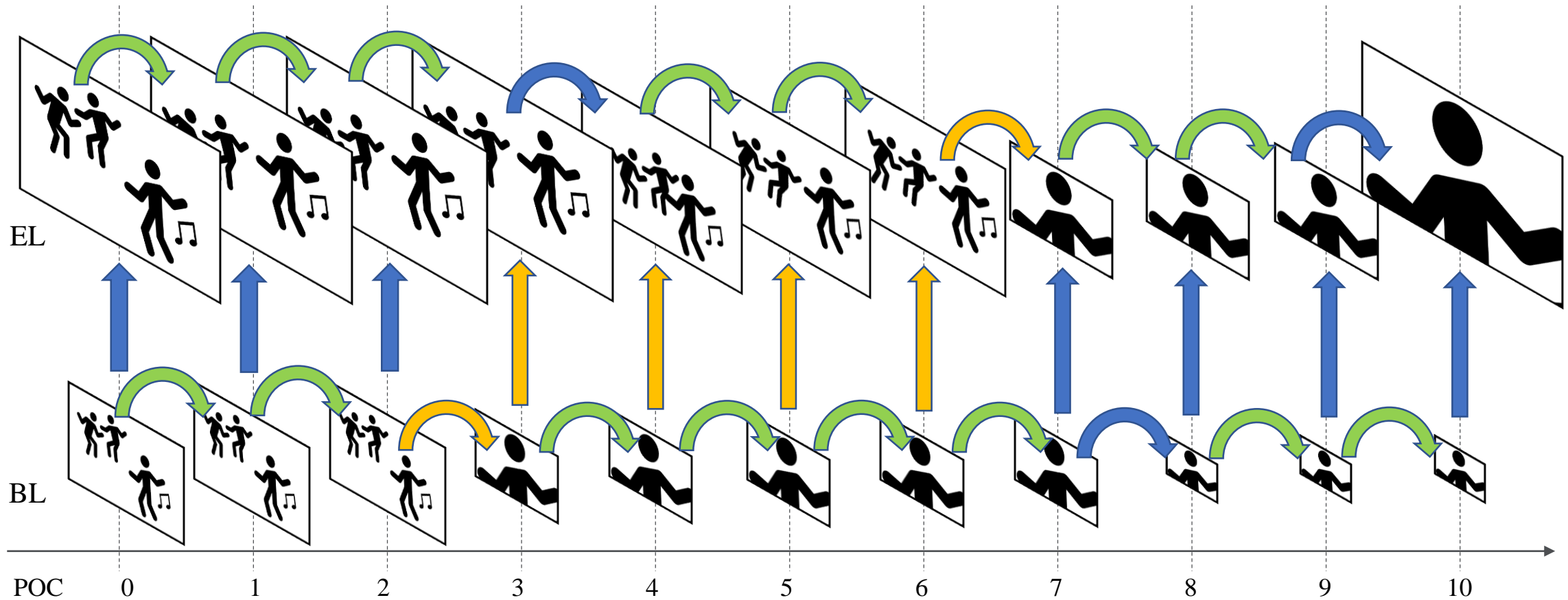
- VVC draft7 supports spatial scalability using RPR framework with BL used as LTRP.
- Adding support for ROI RPR supports ROI scalability automatically when the BL (ROI picture) is completely included in EL full picture.



ROI scalability support

Mixed RPR, ROI RPR, and Spatial Scalability

Green: regular inter prediction Blue: inter prediction with RPR (scaling) Orange: inter prediction with ROI RPR (cropping+ scaling)





Proposed ROI RPR syntax and semantics

ROI Syntax: SPS and Slice Header

Proposal

- Signal ROI window offset parameters as a list in SPS
- Signal index to specific ROI window in slice header

Rationale

- Signaling an ROI window with 4 offsets for every pair of cur pic and ref pic is costly.
- Provides flexibility to have a different ROI window for cur pic or code cur pic using multiple ref pics each with a different ROI window.
- Compared to APS, SPS implementation is simpler when ROI window does not change frequently.

Proposed ROI syntax

ROI window list in sequence parameter set RBSP syntax

	Descriptor
seq_parameter_set_rbsp() {	
...	
if(ref_pic_resampling_enabled_flag) {	
sps_rpr_roi_window_present_flag	u(1)
if(sps_rpr_roi_window_present_flag) {	
num_roi_window_list_minus1	ue(v)
for(i = 0; i <= num_roi_window_list_minus1; i++) {	
roi_offset_prec_shift[i]	ue(v)
roi_offset_list[i][0] /*left */	ue(v)
roi_offset_list[i][1] /*right */	ue(v)
roi_offset_list[i][2] /*top */	ue(v)
roi_offset_list[i][3] /*bottom */	ue(v)
}	
}	
}	

ROI window specification in slice header syntax

	Descriptor
slice_header() {	
...	
if(slice_type != I && sps_rpr_roi_window_present_flag) {	
slice_rpr_roi_window_enabled_flag	u(1)
if(slice_rpr_roi_window_enabled_flag) {	
for(i = 0; i < slice_type == B ? 2: 1; i++) {	
for(j = 0; j < num_ref_entries[i][RplIdx[i]]; j++) {	
curr_roi_window_present_flag[i][j]	u(1)
if(curr_roi_window_present_flag[i][j])	
curr_roi_offset_idx[i][j]	u(v)
ref_roi_window_present_flag[i][j]	u(1)
if(ref_roi_window_present_flag[i][j])	
ref_roi_offset_idx[i][j]	u(v)
}	
}	
}	



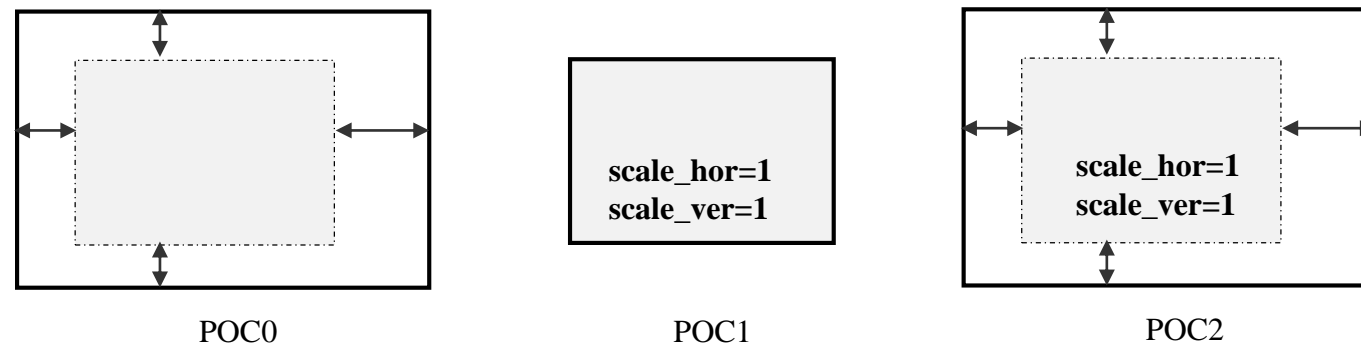
Changes to decoding processes

Specification: Changes to decoding process

- Derivation of scaling ratio
 - 8.3.2 (Decoding process for reference picture lists construction)
 - Add text to construct reference and current ROI window offset parameters
 - Add text of ROI RPR enabled (if not, RPR default)
 - Add additional text for RefPicsScaled to allow scaling ratio equal to 1 if ROI RPR is enabled
- Offset handling in fractional interpolation process
 - 8.5.6.3.1 (Fractional sample interpolation process: General subsection):
 - Add text to specify reference and current ROI window offset parameters as inputs to the process
 - Change variable names for reference picture scale and offsets, and add variables for current picture offsets
 - Change variable names for determination of locations pointed to by motion vector (equations 937, 938, 946, and 948)

Enabling ROI RPR when scaling ratio is equal to 1

- In VVC draft 7, RPR is disabled when ref pic and cur pic are the same size (ScalingRatio=1.0)
- However, ROI RPR is meaningful when ScalingRatio=1.0
 - ROI is created by cropping alone without additional re-sampling
- For ROI RPR, it is proposed to additionally check if the PicOutputWidthL of ref and cur pic or PicOutputHeightL of ref and cur pic is not equal when ScalingRatio=1.0 (which indicates there is an ROI either in cur or ref pic)
 - To determine the applicability of other coding tools (TMVP, BDOF/PROF, DMVP).





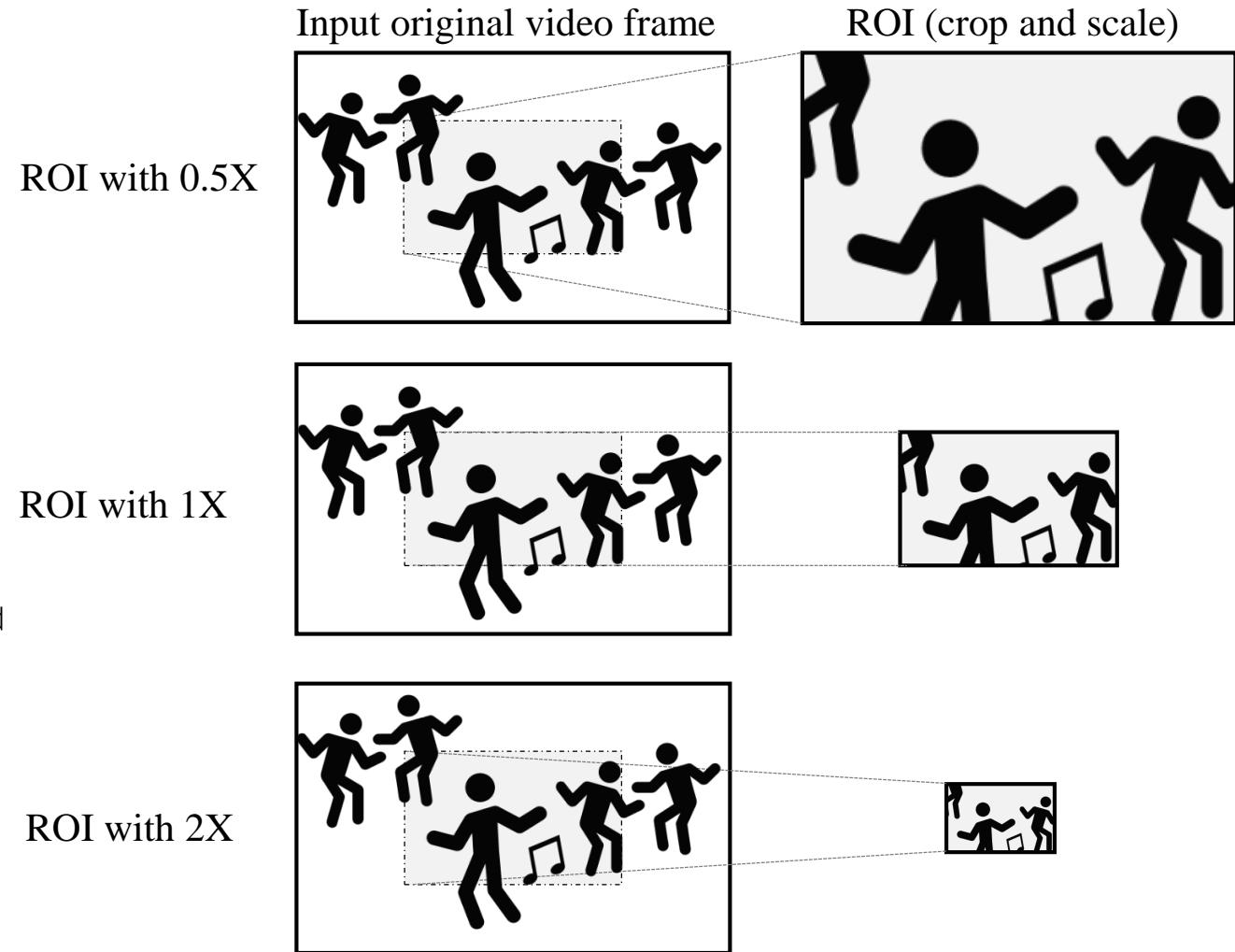
Description of ROI RPR software

Software Implementation

- Based on VTM7.0 and MR1089
 - MR1089: (https://vcgit.hhi.fraunhofer.de/jvet/VVCSsoftware_VTM/merge_requests/1089) JVET-P0590: scaling window implementation and JVET-P0592: chroma phase offset implementation.
 - Thanks Vadim Seregin for providing software
- Changes are mainly modifications of existing RPR framework.
 - Most software changes are made at encoder side.
 - No additional memory storage is introduced.
- Common (Encoder and Decoder)
 - Inter prediction module: fractional sample interpolation process with proper scale and offsets
 - Enable RPR for scale=1X with ROI
- Decoder side only:
 - High level syntax: parsing the ROI window list in SPS and index of ROI window used for each ref in slice header.

Software Implementation: Encoder Pre-processing

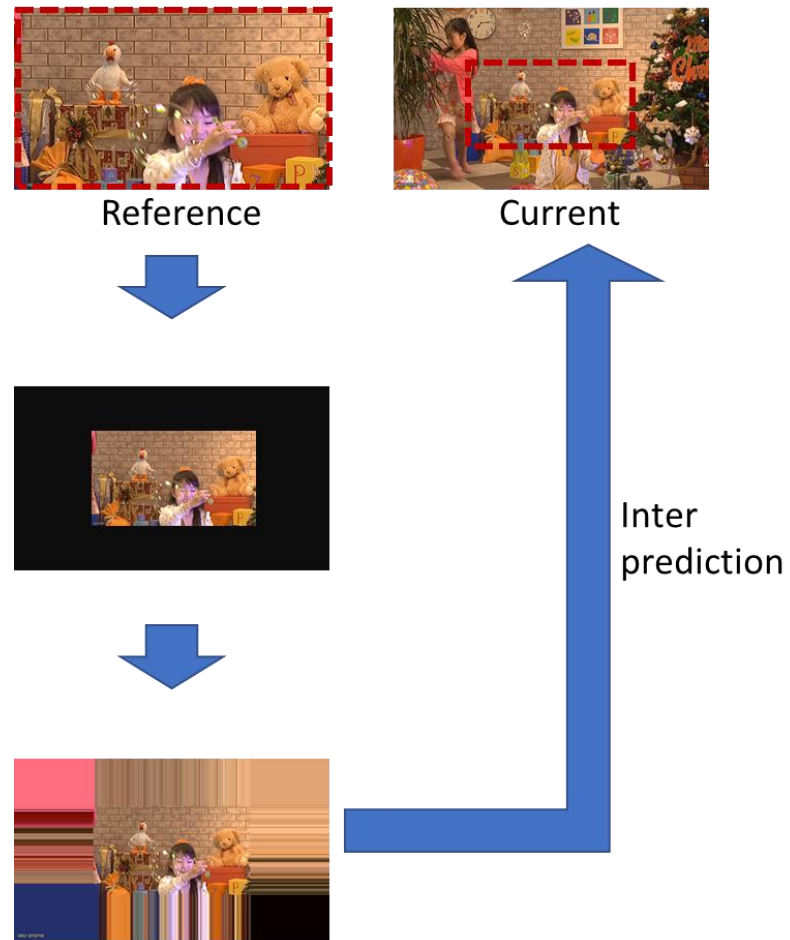
- The RPR pre-processing module (rescale input pictures at a specified period, e.g., every 0.5 second), is modified to add support for cropping.
 - Cropped ROI region can be scaled.
 - Encoder config parameters:
 - position (left and top offset on input picture grid)
 - scale factors (hor_scale and ver_scale)
 - ROI size (multiple of 8)
 - right and bottom ROI offsets are automatically derived based on the other parameters.
 - Supports dumping of pre-processed video (no encoding) to generate non-compressed YUV data for comparison tests.



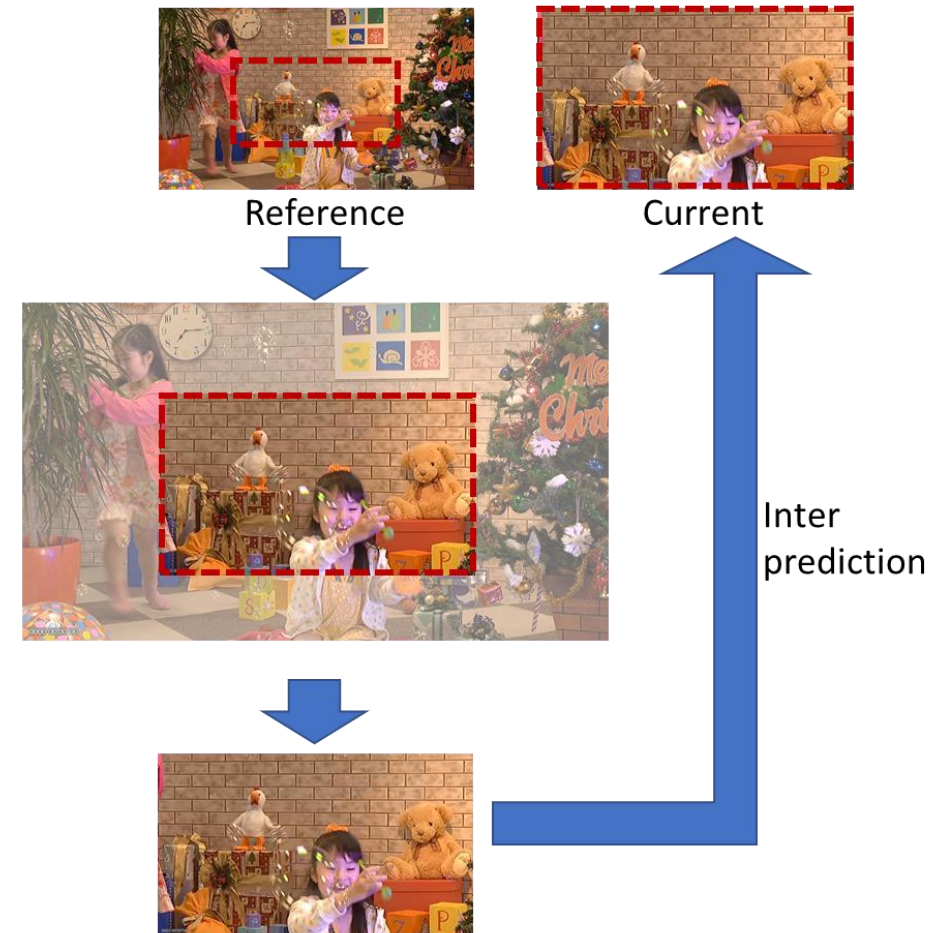
Software Implementation: Encoder Ref Pic Handling

- Ref pic rescaling function is modified with respect to ROI parameters to calculate correct RPR scaling factors and to process each ref pic properly.
 - To reuse regular ME module, scaled ref pic need to have same resolution as cur pic
 - If ref pic corresponds to an ROI in cur pic, scaling with boundary padding to generate scaled ref pic
 - If cur pic corresponds to an ROI in ref pic, scaling with cropping to generate scaled ref pic
- High level syntax: signal the ROI window list in SPS and index of ROI window used for each ref pic in slice header.

Software Implementation



reference picture corresponds to an ROI in current



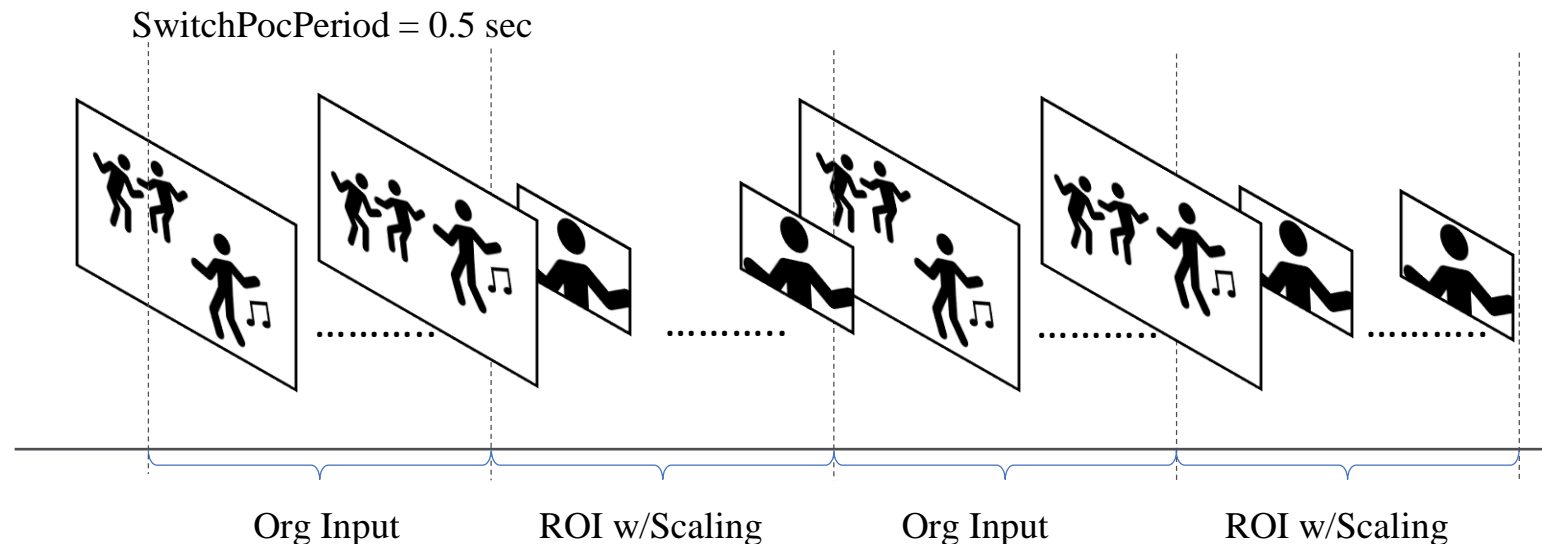
current picture corresponds to an ROI in reference picture



ROI RPR test simulations and results

Experiment Design and Test Condition

- Test condition follows RPR CTC.
 - Switching between original input and ROI happens in every 0.5 seconds
 - FractionNumFrames is set to 0.5 as in RPR tests to encode only half of the whole clip
- Experiment setup
 - ROI is taken as center of picture
 - Functionality: ROI with 3 different scaling factors: 0.5X, 1X, 2X
 - Coding performance: ROI scaling factor 0.5X to have same resolution as input video



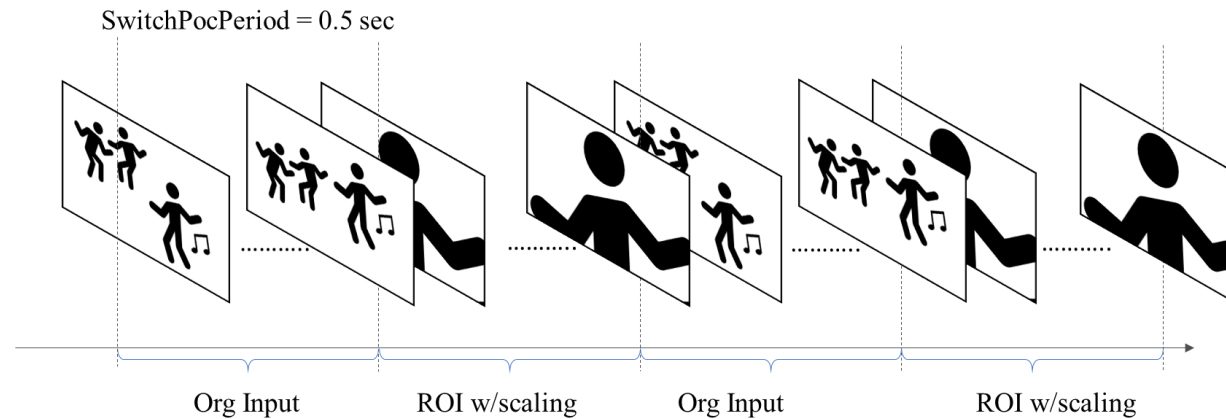
Experiment Design and Test Condition

- Experiment setup: coding performance: ROI 0.5X

Ref:

coding of pre-processed CTC clips with VTM7.0 SW:

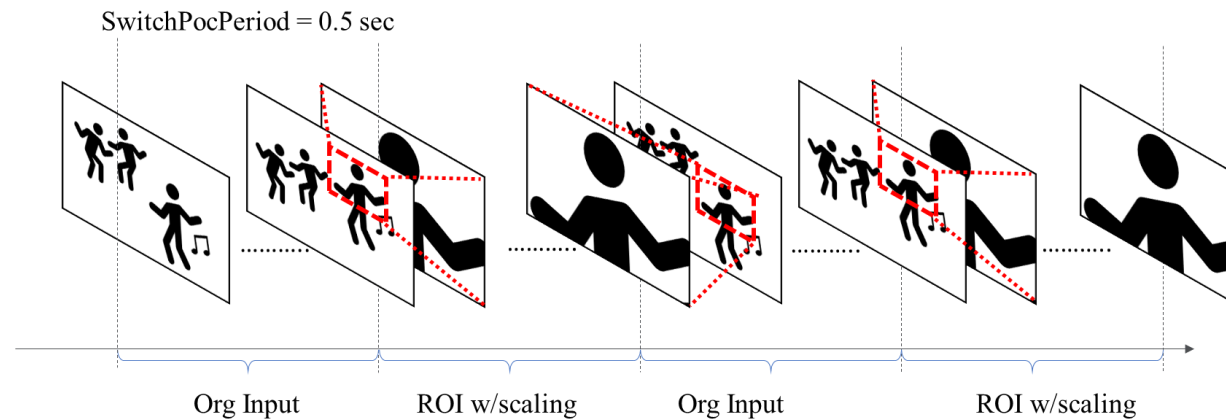
- RPR is not enabled due to same pic resolution
- input video: dumped from automatic ROI cropping/scaling using proposed SW



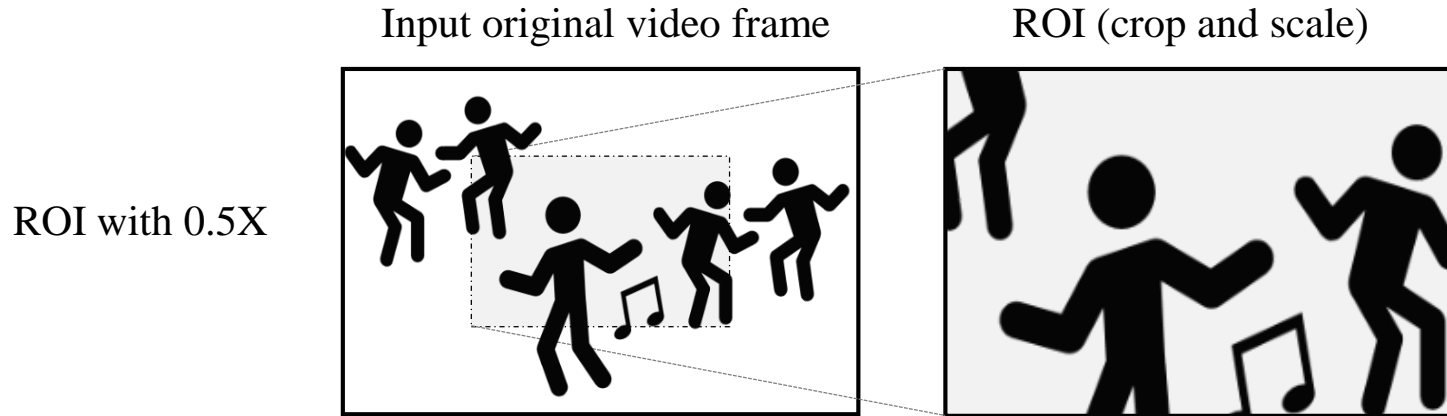
Test (ROI RPR with 0.5X):

coding of input CTC clips with proposed SW:

- automatic ROI cropping/scaling happens before encoding.



Coding Efficiency Test



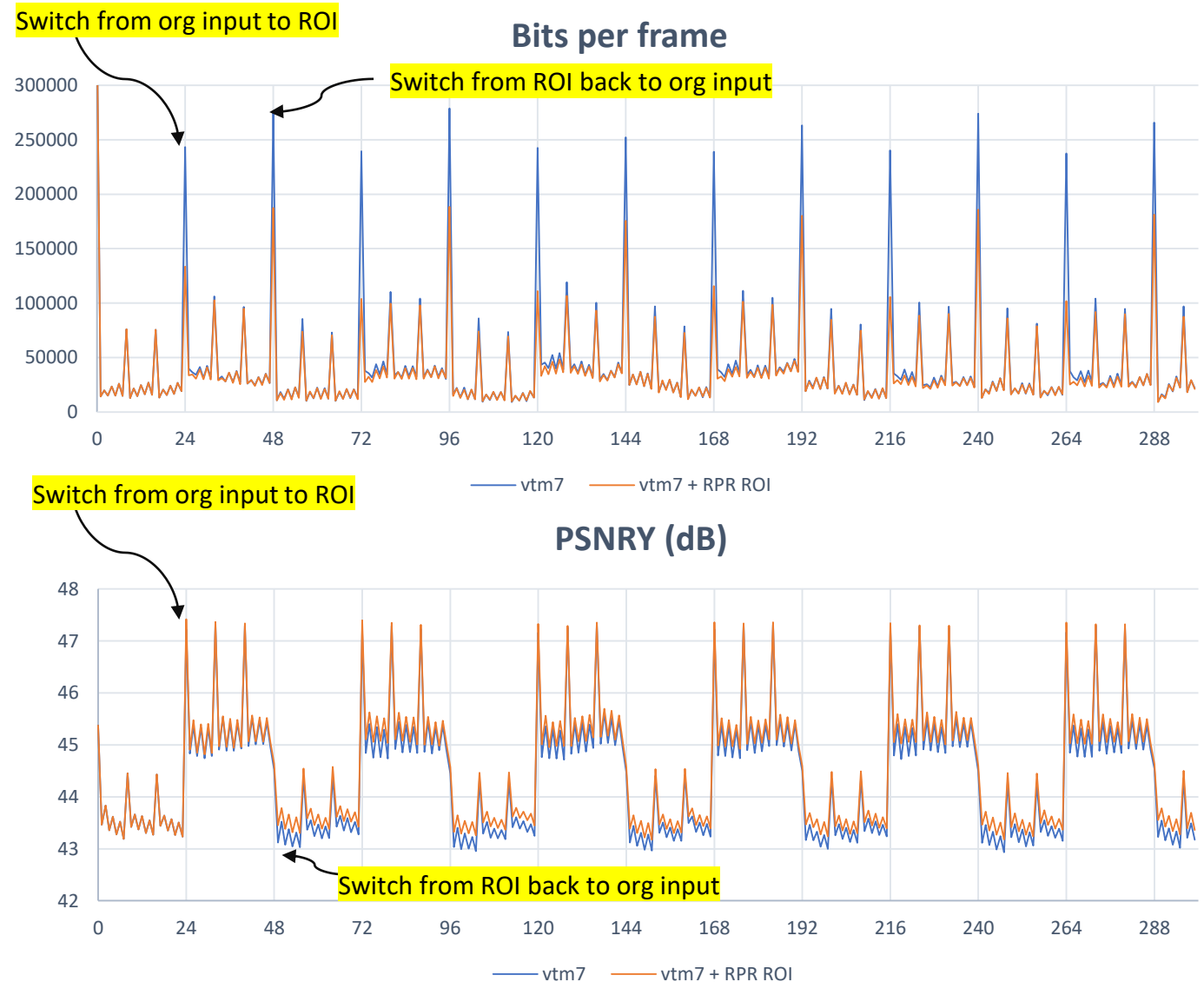
	Low delay B Main10				
	RPR+ROI Over ref				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	-6.94%	-8.58%	-9.00%	107%	102%
Class C	-8.64%	-7.59%	-7.47%	105%	102%
Class E	-21.17%	-22.15%	-20.81%	101%	101%
Overall	-11.06%	-11.65%	-11.44%	105%	102%
Class D	-7.81%	-7.43%	-6.96%	101%	102%
Class F	-16.69%	-15.83%	-15.33%	94%	106%

	Low delay P Main10				
	RPR+ROI Over ref				
	Y	U	V	EncT	DecT
Class A1					
Class A2					
Class B	-5.80%	-6.97%	-7.31%	105%	101%
Class C	-7.91%	-7.43%	-6.48%	105%	101%
Class E	-20.27%	-22.03%	-20.38%	101%	99%
Overall	-10.12%	-10.89%	-10.30%	104%	100%
Class D	-7.21%	-7.28%	-6.83%	107%	101%
Class F	-16.42%	-15.34%	-14.79%	94%	102%

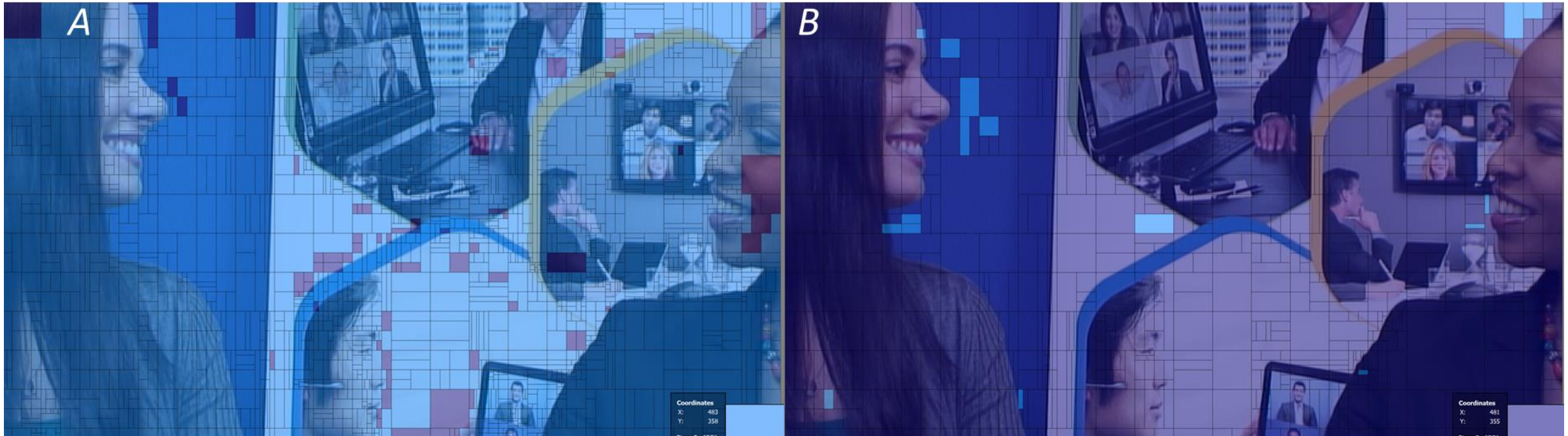
- More than 10% average BDRate saving with proposal
- Suggests that inter prediction being aware of ROI relationship between ref pic and cur pic can improve prediction quality significantly.

Picture-level analysis

- KristenAndSara
 - LDB, QP22
 - Frame 24, 48, 72 etc. are frames where a switching between original input and ROI occurs
 - Switching frames: significant bit savings is achieved with close PSNR



Mode Distribution: KristenAndSara QP22, LDB: POC 24.



VTM7.0

PredMode: 0: INTER (purple); 1: INTRA (blue)

ROI RPR

For POC 24, cur pic is at 0.5 sec switching point:

- Ref pic is an original pic and cur pic (ROI pic) is central ROI of original pic resized to the original pic size.
- ROI RPR (B) uses almost entirely inter prediction modes, whereas VTM 7.0 (A) uses almost entirely intra prediction modes.

Mode Distribution: KristenAndSara QP22, LDB: POC 48.



VTM7.0

PredMode: 0: INTER (purple); 1: INTRA (blue)

ROI RPR

For POC 48, cur pic is at 1.0 sec switching point:

- Ref pic is ROI pic (resized to the orig pic and locate in center of image) and cur pic is original pic.
- ROI RPR test (B) uses more inter prediction modes in central ROI region than VTM 7.0 (A)

Bit Rate Overhead

- ROI window list in SPS:
 - LDB ROI 0.5x: One ROI window
 - Overhead depends on resolution and ROI offsets value
- Slice header bits overhead
 - For CTC LDB configuration (up to 8 ref pics), slice header overhead is 1~25 bits per slice.
 - For CTC LDP configuration (up to 4 ref pics), slice header overhead is 1~13 bits per slice.

Low Delay B Main 10, 0.5X				ROI offsets				
		Width	Height	L	R	T	B	SPS overhead (bits)
Class B	MarketPlace	1920	1080	480	480	270	270	65
	RitualDance	1920	1080	480	480	270	270	65
	Cactus	1920	1080	480	480	270	270	65
	BasketballDrive	1920	1080	480	480	270	270	65
	BQTerrace	1920	1080	480	480	270	270	65
Class C	BasketballDrill	832	480	208	208	120	120	43
	BQMall	832	480	208	208	120	120	43
	PartyScene	832	480	208	208	120	120	43
	RaceHorses	832	480	208	208	120	120	43
Class D	BasketballPass	416	240	104	104	60	60	41
	BQSquare	416	240	104	104	60	60	41
	BlowingBubbles	416	240	104	104	60	60	41
	RaceHorses	416	240	104	104	60	60	41
Class E	FourPeople	1280	720	320	320	180	180	53
	Johnny	1280	720	320	320	180	180	53
	KristenAndSara	1280	720	320	320	180	180	53
Class F	BasketballDrillText	832	480	208	208	120	120	43
	ArenaOfValor	1920	1080	480	480	270	270	65
	SlideEditing	1280	720	320	320	180	180	53
	SlideShow	1280	720	320	320	180	180	53



Memory Bandwidth Constraints

Scaling Ratio Constraints

- The constraints suggested in JVET-Q0331[5] item 1 can be applied for ROI RPR summarized below:
- When $\text{RefPicIsScaled}[i][j]$ is equal to 1, it is a requirement of bitstream conformance that the following constraints apply:
 - $\text{RefPicScale}[i][j][0]$ shall be smaller than or equal to 2^M
 - $\text{RefPicScale}[i][j][0]$ shall be larger than or equal to 2^N
 - $\text{RefPicScale}[i][j][1]$ shall be smaller than or equal to 2^M
 - $\text{RefPicScale}[i][j][1]$ shall be larger than or equal to 2^N

Where $M = 15$, $N = 11$

Worst case MC Memory Constraints

- To prevent worst case MC memory bandwidth increase for supporting RPR, following JVET-Q0179: AHG9: Bitstream conformance requirement related to RPR scaling ratio for worst case MC memory bandwidth reduction

- Using method 1 as example, constraints are rewritten in the context of ROI RPR

$$\text{pic_width_in_luma_samples} * \text{pic_height_in_luma_samples} * (16 * \text{RefPicScale}[i][j][0] / (1 \ll 14) + 7) * (4 * \text{RefPicScale}[i][j][1] / (1 \ll 14) + 7) \leq \text{pic_width_max_in_luma_samples} * \text{pic_height_max_in_luma_samples} * 253$$
$$\text{pic_width_in_luma_samples} * \text{pic_height_in_luma_samples} * (4 * \text{RefPicScale}[i][j][0] / (1 \ll 14) + 7) * (16 * \text{RefPicScale}[i][j][1] / (1 \ll 14) + 7) \leq \text{pic_width_max_in_luma_samples} * \text{pic_height_max_in_luma_samples} * 253$$



Conclusions

Conclusion

- Proposal to add ROI support in RPR
 - Syntax change to add ROI window offsets parameter list in SPS and index to ROI window list for ref pic or cur pic in slice header
 - Decoding process changes are minor: replace scaling window offset parameters with ROI window offset parameters
 - Functionality: extend use cases of RPR and improve coding efficiency for video sequences that contain switching between wide shots and close-up shots of the same scene
 - ROI scalability is automatically supported
- Specification and software implementation are provided
- Propose for adoption

Acknowledgement

- We would like to thank Alibaba for the crosscheck JVET-Q0567
 - Functionality check: ROI RPR 0.5X, 1X, 2X: check encode/decode match
 - Coding efficiency check: VTM7.0 versus ROI RPR 0.5X



JVET-Q0199 AHG8: Support of ROI (Region-of-Interest) RPR

17th JVET Meeting: Brussels, BE, 7–17 January 2020

Dolby Laboratories, Inc.