



JCTVC-D200: High layer syntax to improve support for temporal scalability

Jill Boyce, Danny Hong, Alexandros Eleftheriadis
Vidyo



JCTVC-D200:High layer syntax to improve support for temporal scalability



- **High layer syntax changes are proposed to improve support for temporal scalability in the HEVC design**
 1. Normative semantics for temporal_id
 2. Moving temporal_id_nesting_flag to the sequence parameter set, and adding a temporal_switching_point_flag to the NAL unit header
 3. SEI message describing the temporal picture coding structure

Background

- **H.264/AVC standard is flexible enough to allow for a variety of patterns of coding pictures and reference pictures, even those not initially anticipated**
- **Hierarchical P and hierarchical B coding patterns used for temporal scalability can be supported using `ref_pic_list_modification()` and `dec_ref_pic_marking()`/MMCO syntax elements**
- **SVC extension improved temporal scalability support**
 - Prefix NAL units
 - NAL unit header SVC extension
 - Scalability information SEI messages

1. Normative semantics for temporal_id

- **WD already includes the temporal_id field in the NAL unit header**
- **Propose normative semantics associated with the temporal_id, disallowing reference picture predictions from higher temporal layers**
- **Existing methods**
 - In SVC, temporal_id field is in the NAL unit header SVC extension
 - Required that sub-bitstreams be compliant, but did not have any direct impact on decoding process
 - “All sub-bitstreams that can be derived using the sub-bitstream extraction process as specified in subclause G.8.8.1 with any combination of values for priority_id, temporal_id, dependency_id, or quality_id as the input shall result in a set of coded video sequences, with each coded video sequence conforming to one or more of the profiles specified in Annex A and Annex G.”
 - Ref_pic_list_reordering() can be used to for ref index assignment for hierarchical P and B structures used for temporal scalability with multiple layers

1. Normative semantics for temporal_id

- **Proposal**

- Change the reference picture list initialization process
 - Reference pictures with higher temporal_id values than the temporal_id of the current picture are not assigned indices in the reference picture prediction list
 - Does not change which pictures are stored in the reference picture storage
- Encoders are free to avoid the temporal_id coding restrictions by encoding all pictures with temporal_id equal to 0

- **Text changes for normative semantics for temporal_id**

temporal_id specifies a temporal identifier for the NAL unit. ~~The assignment of values to temporal_id is constrained by the sub-bitstream extraction process as specified in subclause XX.~~

The value of temporal_id shall be the same for all NAL units of an access unit. When an access unit contains any NAL unit with nal_unit_type equal to 5, temporal_id shall be equal to 0.

~~Pictures with higher values of temporal_id can not be used as reference pictures for pictures with lower values of temporal_id, as specified in the decoding process for reference list construction process in subclause X (8.2.4.2.1 and 8.2.4.2.3 in AVC).~~

1. Normative semantics for temporal_id

8.2.4.2.1 Initialisation process for the reference picture list for P and SP slices in frames

This initialisation process is invoked when decoding a P or SP slice in a coded frame.

When this process is invoked, there shall be at least one reference frame or complementary reference field pair that is currently marked as "used for short-term reference" or "used for long-term reference".

Pictures with higher values of temporal_id than the current picture cannot be used for reference, and are not included in the reference picture list. The reference picture list RefPicList0 is ordered so that short-term reference frames and short-term complementary reference field pairs have lower indices than long-term reference frames and long-term complementary reference field pairs.

The short-term reference frames and complementary reference field pairs are ordered starting with the frame or complementary field pair with the highest PicNum value and proceeding through in descending order to the frame or complementary field pair with the lowest PicNum value, **excluding any frame or complementary field pair with a temporal_id value higher than that of the current picture.**

The long-term reference frames and complementary reference field pairs are ordered starting with the frame or complementary field pair with the lowest LongTermPicNum value and proceeding through in ascending order to the frame or complementary field pair with the highest LongTermPicNum value, **excluding any frame or complementary field pair with a temporal_id value higher than that of the current picture.**

2. temporal_id_nesting_flag, temporal_switching_point_flag

- Use of temporal_id alone does not provide enough information for the bitstream extractor to allow switching between temporal layers
- Existing SVC design
 - Includes the temporal_id_nesting_flag in the Scalability information SEI message
 - No decoding process changes

“All sub-bitstreams that can be derived using the sub-bitstream extraction process as specified in subclause G.8.8.1 with any combination of values for priority_id, temporal_id, dependency_id, or quality_id as the input shall result in a set of coded video sequences, with each coded video sequence conforming to one or more of the profiles specified in Annex A and Annex G.”
 - tl_switching_point SEI message to enable temporal layer switching
- Cleaner design will simplify gateway operation

2. temporal_id_nesting_flag, temporal_switching_point_flag

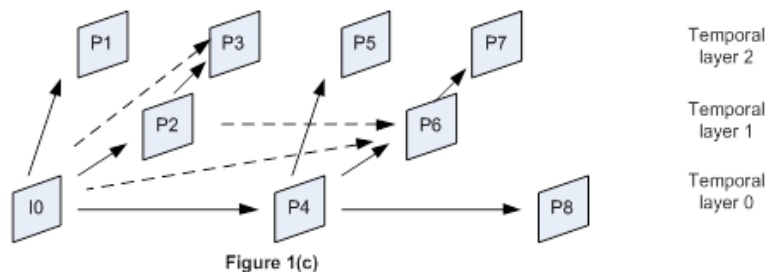
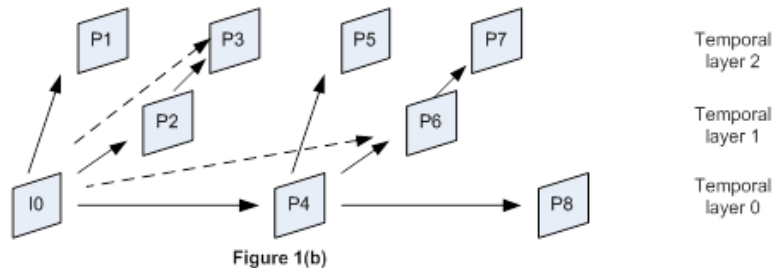
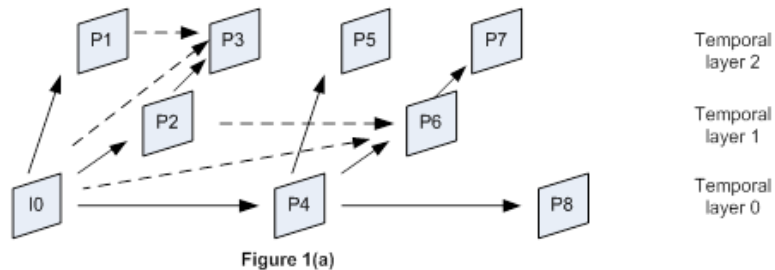
• Proposal

- Move the temporal_id_nesting_flag to the Sequence Parameter Set, and define normative semantics associated with it
 - If temporal_id_nesting_flag is set to 1, when a lower temporal_id picture is decoded, pictures with higher temporal_id values can no longer be used for prediction
 - Enforced in the decoder reference picture marking process by setting as “unused for reference” all pictures in the reference picture list with higher temporal_id values
 - those pictures will be removed from reference picture memory
- Introduce a temporal_switching_point_flag in the NAL unit header
 - More flexibility can be gained by using the temporal_switching_point_flag for individual pictures
 - temporal_id_nesting_flag applies to entire sequence
 - Enables encoders to control the tradeoff between multiple reference picture coding efficiency gains and frequency of temporal switching points

Example: Improved flexibility with per picture temporal_switching_point_flag



Vidyo™
Personal Telepresence



For pictures P3 and P6

- Solid lines first predictor, dashed lines additional multiple ref pictures
- 1(a): temporal_id_nesting_flag is 0
 - P3 predictors: P2, P1, I0
 - P6 predictors: P4, P2, I0
- 1(b) temporal_id_nesting_flag is 1
 - P3 predictors: P2, ~~P1~~, I0
 - P6 predictors: P4, ~~P2~~, I0
- 1(c): temporal_id_nesting_flag is 0, P2's temporal_switching_point_flag is 1, P4's temporal_switching_point_flag is 0
 - P3 predictors: P2, ~~P1~~, I0
 - P6 predictors: P4, P2, I0

Create switching point for layer 1->2 switching, but not layer 0->1 switching

temporal_id_nesting_flag syntax

seq_parameter_set_rbsp() {	C	Descriptor
profile_idc	0	u(8)
reserved_zero_8bits /* equal to 0 */	0	u(8)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
bit_depth_luma_minus8	0	ue(v)
bit_depth_chroma_minus8	0	ue(v)
increased_bit_depth_luma	0	ue(v)
ine_bit_depth_chroma	0	ue(v)
log2_max_frame_num_minus4	0	ue(v)
pic_order_cnt_type	0	ue(v)
if(pic_order_cnt_type == 0)		
log2_max_pic_order_cnt_lsb_minus4	0	ue(v)
else if(pic_order_cnt_type == 1) {		
delta_pic_order_always_zero_flag	0	u(1)
offset_for_non_ref_pic	0	se(v)
num_ref_frames_in_pic_order_cnt_cycle	0	ue(v)
for(i = 0; i < num_ref_frames_in_pic_order_cnt_cycle; i++)		
offset_for_ref_frame[i]	0	se(v)
}		
max_num_ref_frames	0	ue(v)
gaps_in_frame_num_value_allowed_flag	0	u(1)
temporal_id_nesting_flag	0	u(1)

temporal_id_nesting_flag specifies whether inter prediction is additionally restricted for the target access unit set. Dependent on temporal_id_nesting_flag, the following applies.

- If temporal_id_nesting_flag is equal to 0, additional constraints may not be obeyed.
- Otherwise (temporal_id_nesting_flag is equal to 1), the following constraint shall be obeyed for layer representations with any combination of dependency_id and quality_id values present in the target access unit set.

For each access unit auA with temporal_id equal to tldA, an access unit auB with temporal_id equal to tldB and tldB less than or equal to tldA shall not be referenced by inter prediction when there exists an access unit auC with temporal_id equal to tldC and tldC less than tldB, which follows the access unit auB and precedes the access unit auA in decoding order.

NOTE 2 – The syntax element temporal_id_nesting_flag is used to indicate that temporal up-switching, i.e., switching from decoding of up to a specific temporal_id tldN to decoding up to a temporal_id tldM > tldN, is always possible.

Same description as in existing Scalability Information SEI message

temporal_switching_point_flag syntax

nal_unit(NumBytesInNALUnit) {	C	Descriptor
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)
NumBytesInRBSP = 0		
nalUnitHeaderBytes = 1		
if(nal_unit_type == 1 nal_unit_type == 5) {		
temporal_id	All	u(3)
output_flag	All	u(1)
temporal_switching_point_flag	All	u(1)
reserved_zero_3bits	All	u(3)
nalUnitHeaderBytes += 1		
}		

temporal_switching_point_flag specifies if the current access point is a temporal switching point allowing the decoding of higher temporal id layers following this access unit.

If **temporal_switching_point_flag** is equal to 1, all pictures with higher values of temporal_id in the reference picture storage are marked as “unused for reference”, as specified in subclause X.

The value of **temporal_switching_point_flag** shall be the same for all NAL units of an access unit.

If temporal_id_nesting_flag is equal to 1, temporal_switching_point_flag shall be equal to 1.

NOTE – When starting to decode a high temporal layer, availability of required reference pictures can be guaranteed immediately following an IDR, or a picture with a lower value of temporal_id and temporal_switching_flag equal to 1.

temporal_id_nesting_flag, temporal_switching_point_flag semantics



8.2.5 Decoded reference picture marking process

This process is invoked for decoded pictures when `nal_ref_idc` is not equal to 0.

NOTE – The decoding process for gaps in `frame_num` that is specified in subclause 8.2.5.2 may also be invoked when `nal_ref_idc` is equal to 0, as specified in clause 8.

A decoded picture with `nal_ref_idc` not equal to 0, referred to as a reference picture, is marked as “used for short-term reference” or “used for long-term reference”. For a decoded reference frame, both of its fields are marked the same as the frame. For a complementary reference field pair, the pair is marked the same as both of its fields. A picture that is marked as “used for short-term reference” is identified by its `FrameNum` and, when it is a field, by its parity. A picture that is marked as “used for long-term reference” is identified by its `LongTermFrameIdx` and, when it is a field, by its parity.

Frames or complementary field pairs marked as “used for short-term reference” or as “used for long-term reference” can be used as a reference for inter prediction when decoding a frame until the frame, the complementary field pair, or one of its constituent fields is marked as “unused for reference”. A field marked as “used for short-term reference” or as “used for long-term reference” can be used as a reference for inter prediction when decoding a field until marked as “unused for reference”.

A picture can be marked as “unused for reference” by the sliding window reference picture marking process, a first-in, first-out mechanism specified in subclause 8.2.5.3 or by the adaptive memory control reference picture marking process, a customised adaptive marking operation specified in subclause 8.2.5.4, **or by considering the temporal layer prediction restrictions from temporal_switching_point_flag and temporal_id_nesting_flag.**

A short-term reference picture is identified for use in the decoding process by its variables `FrameNum` and `FrameNumWrap` and its picture number `PicNum`, and a long-term reference picture is identified for use in the decoding process by its long-term picture number `LongTermPicNum`. When the current picture is not an IDR picture, subclause 8.2.4.1 is invoked to specify the assignment of the variables `FrameNum`, `FrameNumWrap`, `PicNum` and `LongTermPicNu`

temporal_id_nesting_flag, temporal_switching_point_flag semantics



8.2.5.1 Sequence of operations for decoded reference picture marking process

Decoded reference picture marking proceeds in the following ordered steps.

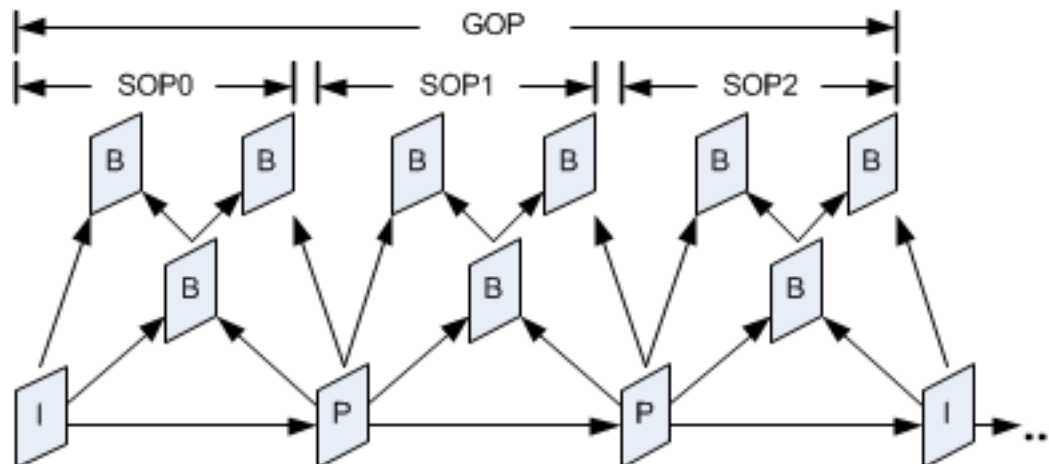
1. All slices of the current picture are decoded.
2. Depending on whether the current picture is an IDR picture, the following applies.
 - If the current picture is an IDR picture, the following applies.
 - All reference pictures are marked as "unused for reference".
 - Depending on long_term_reference_flag, the following applies.
 - If long_term_reference_flag is equal to 0, the IDR picture is marked as "used for short-term reference" and MaxLongTermFrameldx is set equal to "no long-term frame indices".
 - Otherwise (long_term_reference_flag is equal to 1), the IDR picture is marked as "used for long-term reference", the LongTermFrameldx for the IDR picture is set equal to 0, and MaxLongTermFrameldx is set equal to 0.
 - Otherwise (the current picture is not an IDR picture), the following applies.
 - If temporal_switching_point_flag is equal to 1 or temporal_id_nesting_flag is equal to 1, all reference pictures with value of temporal_id greater than the current temporal_id are marked as "unused for reference"
 - If adaptive_ref_pic_marking_mode_flag is equal to 0, the process specified in subclause 8.2.5.3 is invoked.
 - Otherwise (adaptive_ref_pic_marking_mode_flag is equal to 1), the process specified in subclause 8.2.5.4 is invoked.

3. SEI message describing the temporal picture coding structure

- **Current HEVC design and the H.264/AVC standard allow for flexible patterns of coded pictures**
- **However, many encoders, including JM and HM reference software, generate sequences of coded pictures following a fixed pattern**
- **Even when a fixed pattern is in used, gateways and decoders are not aware of it**
- **Propose sending SEI message with explicit coded picture pattern information**
- **Knowing the coding structure a priori can help in dropping frames by gateways**
 - Transraters can optimally choose frames to transcode and minimize drift
 - DVRs can use for fast forward, reverse
 - Parallel decoding simplified and improved when dependencies known

3. SEI message describing the temporal picture coding structure

- GOP (Group of Pictures) is defined as a sequence of coded pictures that start with an I/IDR picture and ends with a picture immediately preceding the next I/IDR picture
- SOP (Structure of Pictures) is smaller repeating pattern, beginning with a first temporal layer picture followed by the repeating set of pictures
- GOP starts with I frame, contains 1 or more SOPs



3. SEI message describing the temporal picture coding structure

coding_structure(payloadSize) {	Descriptor
num_pictures_in_sop_minus1	ue(v)
num_sops_in_gop	ue(v)
for(i = 0; i < num_pictures_in_sop_minus1; i++) {	
primary_pic_type[i]	u(2)
ref_flag[i]	u(1)
temporal_num[i]	u(3)
display_num[i]	ue(v)
}	
average_frame_rate_flag	u(1)
average_bit_rate_flag	u(1)
if(average_frame_rate_flag)	
average_frame_rate	u(16)
if(average_bit_rate_flag)	
for(i = 0; i < NumTemporalLayers; i++)	
average_bit_rate[i]	u(16)
}	

num_pictures_in_sop_minus1 specifies the number of pictures in the SOP minus 1. In the case of all intra or IPPP coding, this value shall be 0.

num_sops_in_gop specifies the number of SOPs in a GOP (between two intras). In the case of all intra coding, this value shall be 1. **num_sops_in_gop** equal to 0 specifies that there is no specific GOP structure and that the next expected I picture is unknown, or that the coded video sequence consists of just one GOP.

primary_pic_type has the same definition as specified in Table 7-2 of WD1 of HEVC [2].

ref_flag equal to 1 specifies that the coded picture is a reference picture. **ref_flag** equal to 0 specifies that the coded picture is not a reference picture.

temporal_num specifies the temporal_id value associated with the coded picture.

display_num specifies the display order of the coded picture within the SOP. Except for the very first picture in the SOP, each picture in the SOP is described in the coding order, which may be different from the display order. This unique number specifies the display number of the coded picture within the SOP.

average_frame_rate_flag specifies whether the average frame rate (**average_frame_rate**) is specified.

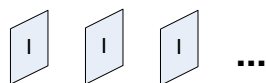
average_frame_rate specifies the average frame rate in units of frames per 256 seconds of the entire coded sequence. Using this information and the **temporal_num** information gathered for each coded picture in the SOP, the frame rate of each temporal layer can be derived.

average_bit_rate_flag specifies whether the average bit rate (**average_bit_rate**) is specified.

average_bit_rate indicates the average bit rate in units of 1000 bits per second of the temporal layer i. All NAL units of the temporal layer, including all NAL units of the temporal layer j, where j < i, are taken into account in the calculation.

Examples

3.1 All intra

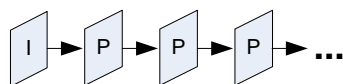


num_pictures_in_sop_minus1: 0

num_sops_in_gop: 1

NumTemporalLayers is derived to be 1

3.2 IPPP

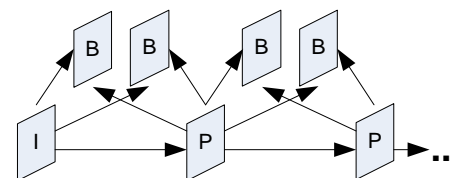


num_pictures_in_sop_minus1: 0

num_sops_in_gop: 0

NumTemporalLayers is derived to be 1

3.3 IBBPBBPBBPBI... (Open GOP)



num_pictures_in_sop_minus1: 2

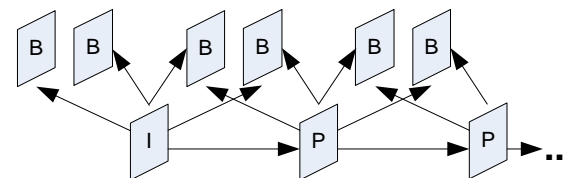
num_sops_in_gop: 4

i = 0: (2 (B), 0, 1, 1)

i = 1: (2 (B), 0, 1, 2)

NumTemporalLayers is derived to be 2

3.4 IBBPBBPBBPBI... (Closed GOP)



num_pictures_in_sop_minus1: 2

num_sops_in_gop: 4

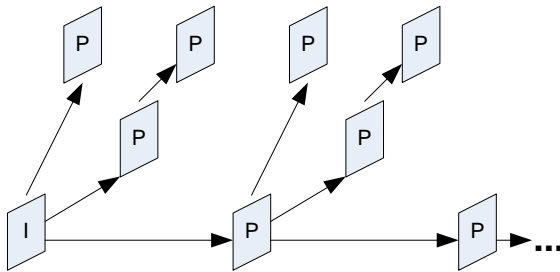
i = 0: (2 (B), 0, 1, 0)

i = 1: (2 (B), 0, 1, 1)

NumTemporalLayers is derived to be 2

Examples

3.5 Hierarchical-P with 3 temporal layers



num_pictures_in_sop_minus1: 3

num_sops_in gop: 0

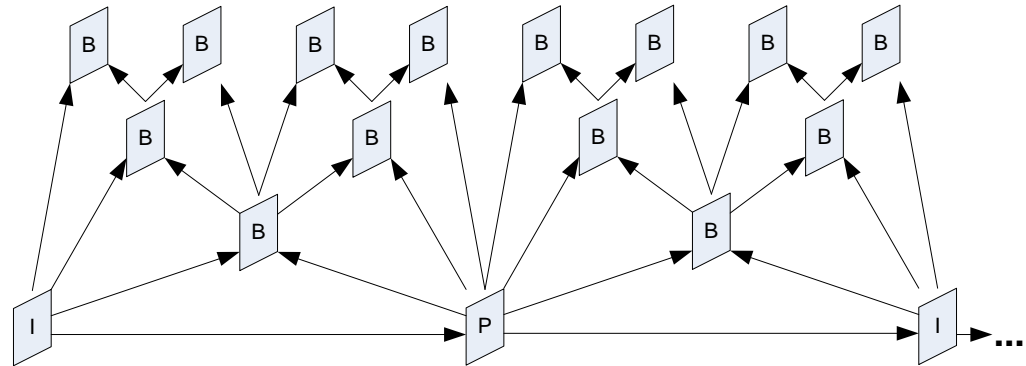
i = 0: (1 (P), 0, 2, 1)

i = 1: (1 (P), 1, 1, 2)

i = 2: (1 (P), 0, 2, 3)

NumTemporalLayers is derived to be 3

3.6 Hierarchical-B with 4 temporal layers



num_pictures_in_sop_minus1: 7

num_sops_in gop: 2

i = 0: (2 (B), 1, 1, 4)

$$i = 1: (2 \text{ (B)}, 1, 2, 2)$$

i = 2: (2 (B), 0, 3, 1)

$$i = 3: (2 \text{ (B)}, 0, 3, 3)$$

i = 4: (2 (B), 1, 2, 6)

i = 5: (2 (B), 0, 3, 5)

i = 6: (2 (B), 0, 3, 7)

NumTemporalLayers is derived to be 4

Conclusions

- **Temporal scalability is widely adopted and accepted**
 - Used in the HEVC test conditions
 - Typically improves coding efficiency
- **Cleaner design possible for HEVC than available in AVC and SVC**
 - Not restricted by backwards compatibility
- **Various types of bitstream extractors and decoders can benefit from improved support for temporal scalability**
 - Gateways with temporal layer switching
 - PVRs
 - Transraters
 - Parallel decoders