Sequence Adaptive Field/Frame practice for fixed-output rate

Harmonic & Ambarella

JCTVC-K0160v1

What is being proposed

- **Cadence** metadata added to frame sequence pictures to drive SAFF output process.
  - Smooth transitions across field/frame sequences
  - Separate, but compatible with HEVC spec's non-normative HRD.
- **Padding** metadata (frame_display_pattern) instructs SAFF output model how to pad output frame sequences with specific repeated pictures for fixed frame rate systems.
  - No padding for field sequences (continue to marked coded repeat field pictures with duplicate_flag)
- **Pairing** association pointers for field sequence pictures (field_display_pattern) so display process knows explicitly which progressive fields to weave together for progressive display output.

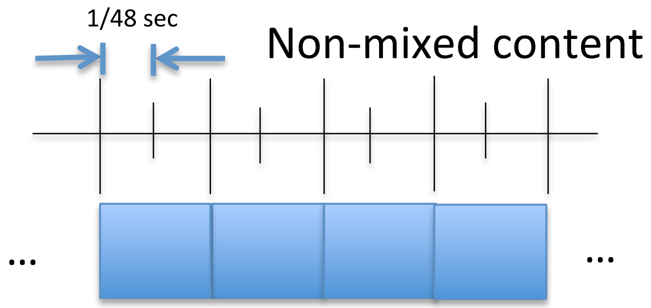
Goals of this tutorial

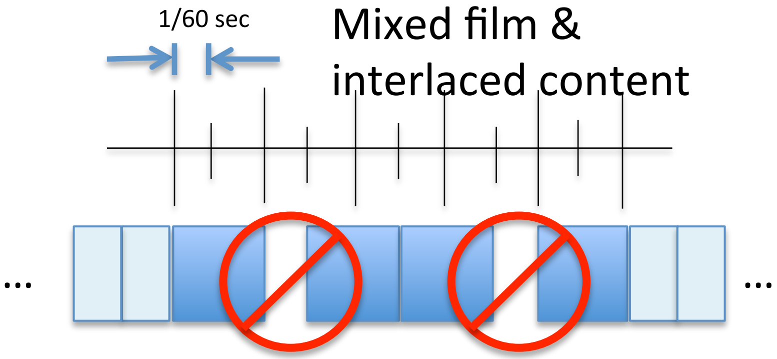
- Smooth broadcast practice transition from MPEG-2 & AVC to HEVC
- Use metadata to deterministically guide output to concurrent interlace and progressive display models
  - How-to convey explicit field cadence for traditional interlace display targets (480i, 576i, 1080i)
- Define clear behavior at SAFF transitions
- Encourage practice of paired field sequence pictures

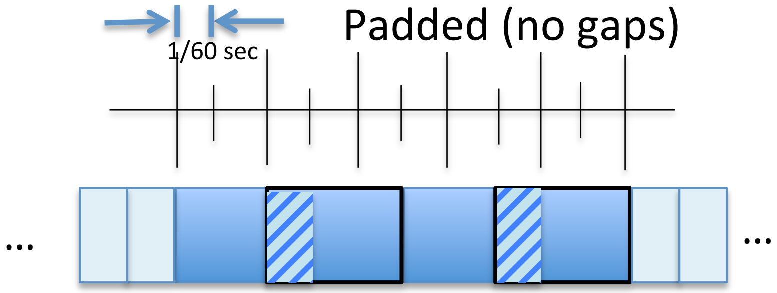
Premise

- HEVC broadcast encoders continue to produce fixed_pic_rate_flag=1 bitstreams as per AVC & MPEG-2 for DVB, SCTE, etc.
- Field or frame sequence type (field_seq_flag) is selected based on what is most efficient for compression.
- Progressive or interlace content is independently identified per picture (progressive_source_flag) to aid display process
- Only progressive content (progressive_source_flag=1) in frame sequences (field_seq_flag=0) would have repeat fields.
  - display target frame rates > 30 Hz repeat entire frames, not fields.
- Interlace content (progressive_source_flag=0) never repeats pictures, and should be encoded in field sequences, though it is possible to encoded as frame sequences and guide proper display output behavior.
- All pics within the a sequence are of the same struct type (field or frame)
  - A new sequence & IDR must be started when switching between fields/frames
- Bitstreams constructed for progressive display targets (720p60) consist only of progressive content; use only frame repetition metadata (duplicate_flag and frame_display_pattern) while
  - field_pic_flag always ‘0’
  - progressive_source_flag always ‘1’

Fixed vs. non-fixed picture rates







1, 1½, 2, or 3 frame intervals
(never less than 1)

Minimalist variable-frame-rate solution:

- At beginning of frame sequence, indicate the display output pattern of the first output frame; let the decoder derive cadence and timing for all frames that follow until the next sequence (unless overridden by PTS and dpb_output_delay).

For fixed frame-rate applications:

- leave no empty picture slots in the *output* sequence.
- Streams must keep display cadence
- Do not switch frame rate more than once per second

A big benefit of padding metadata is that displays can recover the embedded ~24 fps film content from a ~60 Hz output sequence.

Application Restrictions

- In DVB TS 101 154 v 1.10.1 Section 5.5.3.4:
  - *In the case of still picture the **fixed_frame_rate_flag** shall be equal to 0. In other cases, the **fixed_frame_rate_flag** shall be equal to 1.*
  - The frame rate can not be changed between two IDR access units.

- In SCTE 128 2010-a Table 7
  - fixed_frame_rate_flag 1 (equals 0 for Low Delay mode and still pictures)

- HEVC section C.5.2 Removal of pictures from the DPB
  - When the IDR or BLA picture is not the first picture decoded and [..] pic_height_in_luma_samples [..] from the active sequence parameter set is different from the [..] set that was active for the preceding picture, no_output_of_prior_pics_flag is inferred to be equal to 1 by the HRD, regardless of the actual value of no_output_of_prior_pics_flag.
  - NOTE – Decoder implementations should try to handle picture or DPB size changes more gracefully than the HRD [..]
  - When no_output_of_prior_pics_flag is equal to 1 or is inferred to be equal to 1, all picture storage buffers in the DPB are emptied without output of the pictures they contain.

- SCTE 128 section 7.2.1.1 Sequence Parameter Set (SPS) constraints

  "The time interval between consecutive changes in pairs of pic_width_in_mbs_minus1 and pic_height_in_map_units_minus1 shall be greater than or equal to one second."

NOTE: unlike AVC, there is no HEVC field-pair concept

From AVC section C.4.5.3. (Bumping process):

The picture or complementary reference field pair that is first for output is selected as follows:

The frame buffer is selected that contains the picture having the smallest value of PicOrderCnt( ) of all pictures in the DPB marked as "needed for output".

Depending on the frame buffer, the following applies:

If this frame buffer contains a complementary non-reference field pair with both fields marked as "needed for output" and both fields have the same PicOrderCnt( ), the first of these two fields in decoding order is considered first for output.

Otherwise, if this frame buffer contains a complementary reference field pair with both fields marked as "needed for output" and both fields have the same PicOrderCnt( ), the entire complementary reference field pair is considered first for output.

NOTE – When the two fields of a complementary reference field pair have the same value of PicOrderCnt( ), this "bumping" process will output these pictures together, although the two fields have different output times from a decoder that satisfies output timing conformance criteria (as specified in subclause C.2.2).

Otherwise, the picture in this frame buffer that has the smallest value of PicOrderCnt( ) is considered first for output.

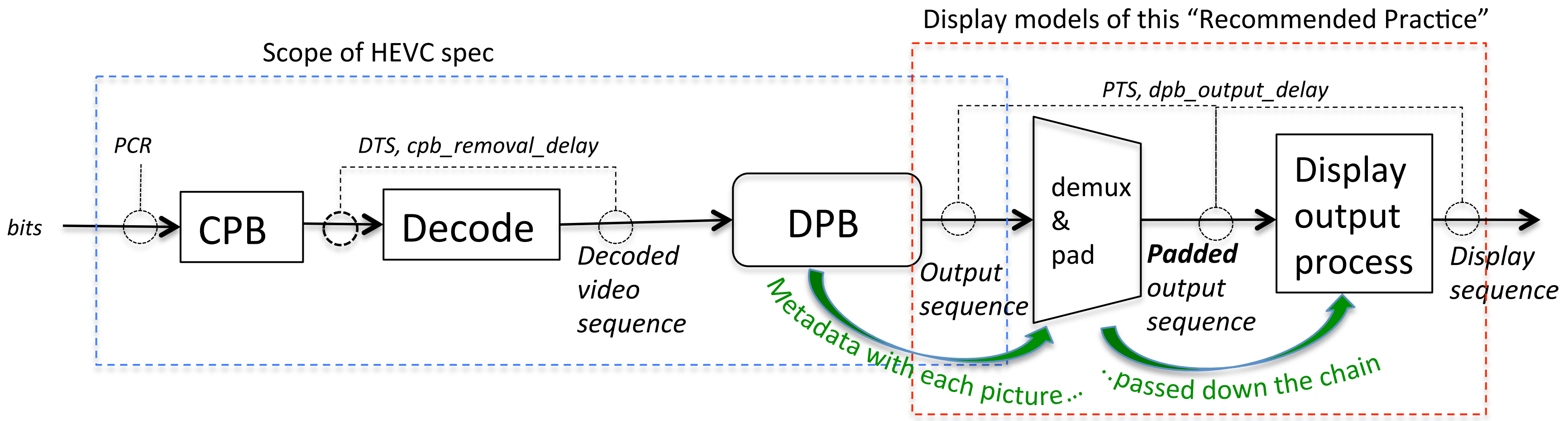
Section 8.2.4.3 of AVC (reference picture list initialization):

NOTE 2 – A non-paired reference field is not used for inter prediction of frames (independent of the value of MbaffFrameFlag).

SCTE 128 section 7.2.1.3 Picture Parameter Constraints and Level Limits

AVC Bitstreams shall not include non-paired fields (as defined in AVC).

Domains



Decoding process: produces alternating field- and frame- decoded video sequences. Display_output_indication() SEI does not affect video coding layer. Encoder calculates Decode timing (DTS, cbp_removal, etc.) according to padded output sequence.

Output: no changes to HEVC practice. Encoder calculates output timing and order (PTS, dpb_output_delay, POC) w.r.t. padded output sequence.

Display: *_display_pattern informs how to construct padded display sequence from padded output sequence and metadata. Same timing as padded output sequence. (fields become frames, visa versa, or no change)

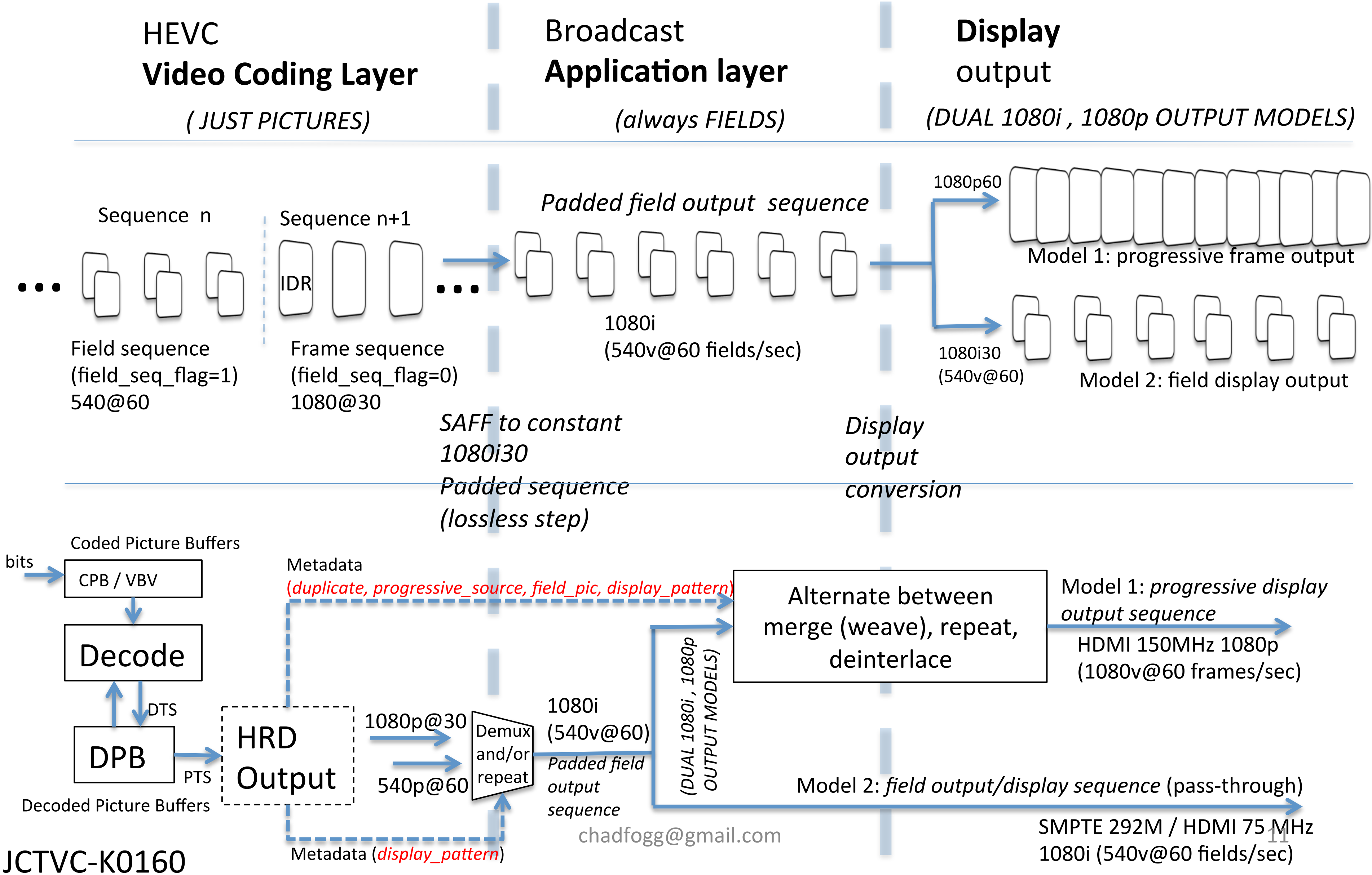
Field sequence

- 1080i is passed through HEVC chain as 540v@60 pps (pictures per second) as far as the Video Coding Layer (VCL) is concerned.
- Metadata in display_output_indication() SEI informs how field sequence is a mixture of interlace (progressive_source_flag=0), progressive field-pairs (progressive_source_flag=1, field_pair_ptr), and 3:2-like film content (alternating duplicate_flag= 0 or 1).
- Any repeated fields would be coded with almost all skipped macroblocks referencing the field being repeated, but marked by duplicate_flag=1.

Frame sequence

- For level <= 4.1, a 30 Hz frame sequence is, and appears, as 1080v@30pps (pictures per second) sequence through VCL.
- Fixed-frame-rate television applications (SCTE, DVB, etc.) will see the output as 1080i (540v@60) demultiplexed from 1080v@30 for sequence padding purposes and require NumClockTS ~1/60 sec tick increments of 1,2 & 3 (e.g. time_scale = 60000, num_units_in_tick=1001).
- Variable frame rate for, e.g, 24 fps film, would signal 24 fps directly (as per Blu-Ray option), or ~30 fps with dpb_delay's alternating ~2/60 and ~3/60.
- 60 frame/sec progressive frame sequence (720p60) will use "frame doubling" and "frame tripling"; time_scale=120000 in DVB, SCTE, etc.

Domains example (1080i)



HEVC
Video Coding Layer
(JUST PICTURES)

Broadcast
Application layer
(always FIELDS)

Display
output
(DUAL 1080i , 1080p OUTPUT MODELS)

Sequence n

Field sequence
(field_seq_flag=1)
540@60

Sequence n+1

IDR

Frame sequence
(field_seq_flag=0)
1080@30

Padded field output sequence

1080i
(540v@60 fields/sec)

1080p60

Model 1: progressive frame output

1080i30
(540v@60)

Model 2: field display output

SAFF to constant 1080i30 Padded sequence (lossless step)

Display output conversion

bits

Coded Picture Buffers

CPB / VBV

Decode

DTS

DPB

PTS

Decoded Picture Buffers

HRD Output

Metadata
(*duplicate, progressive_source, field_pic, display_pattern*)

1080p@30

540p@60

Demux and/or repeat

Metadata (*display_pattern*)

1080i
(540v@60)
Padded field output sequence

(DUAL 1080i , 1080p OUTPUT MODELS)

Alternate between merge (weave), repeat, deinterlace

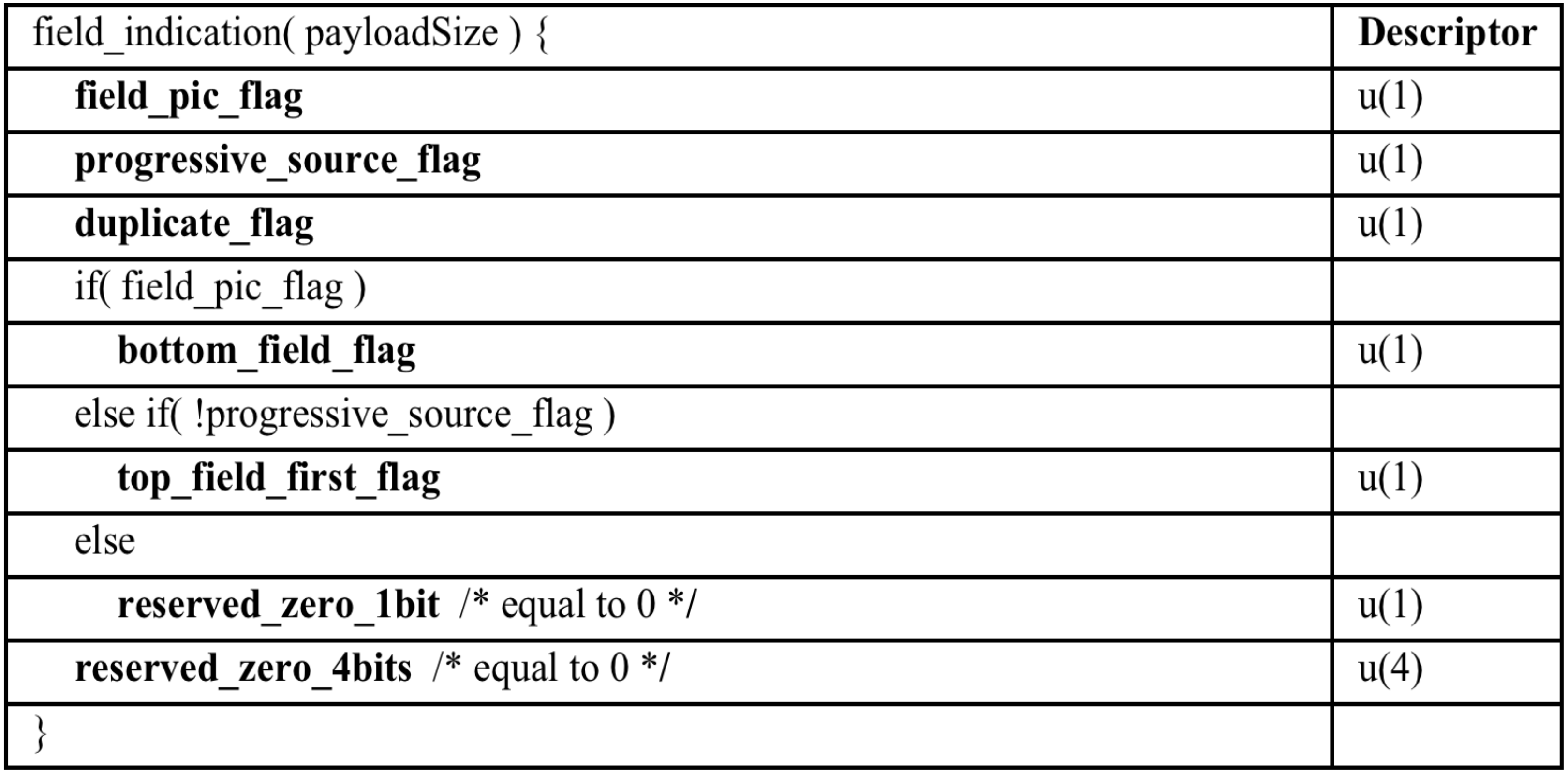
Model 1: *progressive display output sequence*
HDMI 150MHz 1080p
(1080v@60 frames/sec)

Model 2: *field output/display sequence* (pass-through)
SMPTE 292M / HDMI 75 MHz
1080i (540v@60 fields/sec)

Metadata philosophy

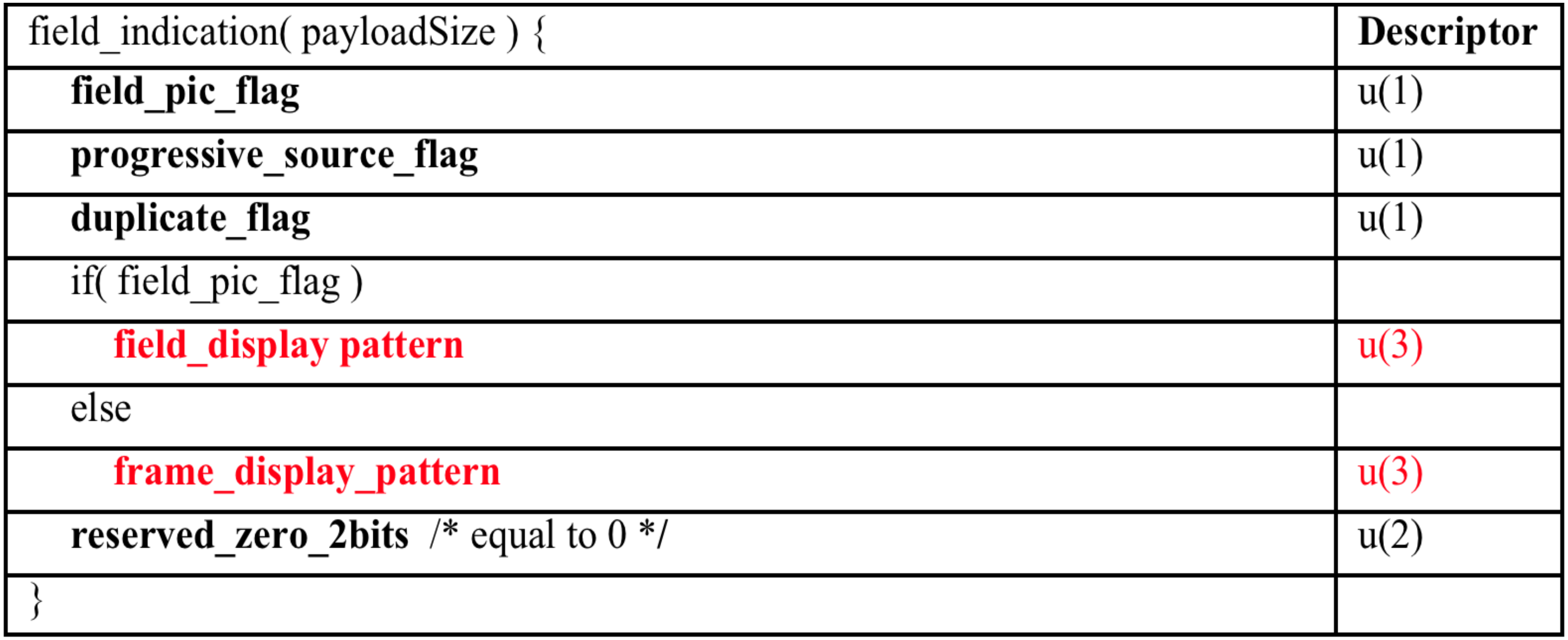
- Metadata in display_output_indication() SEI passes through chain with each coded picture (access unit)
- Metadata is constructed so that there are no gaps in both 1080i (540v@60field/sec) and 1080p (1080v@60 frames/sec) that the display process must guess how to fill (temporally pad sequence)
- How content is coded (field_seq_flag) is independent of content type (progressive_source_flag)

Current metadata syntax



field_indication(payloadSize) {	**Descriptor**
field_pic_flag	u(1)
progressive_source_flag	u(1)
duplicate_flag	u(1)
if(field_pic_flag)	
bottom_field_flag	u(1)
else if(!progressive_source_flag)	
top_field_first_flag	u(1)
else	
reserved_zero_1bit /* equal to 0 */	u(1)
reserved_zero_4bits /* equal to 0 */	u(4)
}	

Proposed field indication change



field_indication(payloadSize) {	**Descriptor**
field_pic_flag	u(1)
progressive_source_flag	u(1)
duplicate_flag	u(1)
if(field_pic_flag)	
field_display pattern	u(3)
else	
frame_display_pattern	u(3)
reserved_zero_2bits /* equal to 0 */	u(2)
}	

Also suggest changing name from *field_indication* to *display_output_indication*

Metadata elements

Element	Semantics
field_seq_flag *from: vui_parameters()*	0: frame sequence *(default)* 1: field sequence. *display_output_indication() in all access units*
field_pic_flag	Per-picture reminder of sequence structure type. *Required to be same value as field_seq_flag*
progressive_source_flag	0: picture is comprised of interlace content 1: picture is comprised of progressive content *(default)*
duplicate_flag	Hint to display, splicers, etc. that picture is a coded duplicate in order to pad the sequence for constant frame rate operation
field_display_pattern	*(new)* Indicates current picture parity and which (if any) field it is associated with in a pair.
frame_display_pattern	*(new)* repetition count/field repeat order (default: no repetition)

field_display_pattern

value	Indicated display
0	Top field paired with previous bottom field in output order
1	Bottom field paired with previous top field in output order
2	Top field paired with next field in output order
3	Bottom field paired with next field in output order
4	Top unpaired field
5	Bottom unpaired field.
6	Reserved
7	Reserved

frame_display_pattern

value	Display output order	Notes	NumClockTS
0	progressive frame	progressive_source_flag=1	1
1	top field, then bottom field		2
2	bottom field, then top field		2
3	top field, then bottom field, then repeat top field	progressive_source_flag=1	3
4	bottom field, then top field, then repeat bottom field	progressive_source_flag=1	3
5	frame doubling	progressive_source_flag=1	2
6	frame tripling	progressive_source_flag=1	3
7	reserved		

pic_struct alternative

- Create pic_struct SEI message or re-introduce pic_struct in pic_timing() SEI
  - Include clock_timestamp_flag[] to signal ct_type and full_timestamp_flag{} elements to signal field pairs that belong to the same progressive frame time
- Essentially eliminates need for display_output_indication SEI
  - Continuity with AVC practice, but might confuse reader into thinking PAFF is in the toolkit.
  - still need **field_seq_flag** in VUI (sequence layer) to inform decoder whether sequence is field or frame.
  - duplicate_flag needed per coded picture when field_seq_flag=1

pic_struct syntax

```
If( pic_struct_present_flag ){
    pic_struct
    for(i=0;i<NumClockTS;i++){
        if( clock_timestamp_flag[i]){
            ct_type
            [..]
            n_frames
            [..] hours, minutes, seconds
        }
}
```

Essentially copied from AVC pic_timing() syntax (§D.1.2) and semantics (§D.2.2)

pic_struct

Adapted from AVC Table D-1

Value	Indicated display order	Restrictions	NumClockTS
0	(Progressive) frame	field_seq_flag shall be 0 (frame sequence) ct_type shall be 0 (progressive)	1
1	Top field	field_seq_flag shall be 1	1
2	Bottom field	field_seq_flag shall be 1	1
3	Top field, then bottom field	field_seq_flag shall be 0	2
4	Bottom field, then top field	field_seq_flag shall be 0	2
5	Top field, then bottom field, then the repeat top field	field_seq_flag shall be 0 ct_type shall be 0	3
6	Bottom field, then top, then repeat bottom field	field_seq_flag shall be 0 ct_type shall be 0	3
7	Frame doubling	field_seq_flag shall be 0 ct_type shall be 0 fixed_pic_rate_flag shall be 1 field_display_target shall be 0 (progressive)	2
8	Frame tripling	field_seq_flag shall be 0 ct_type shall be 0 fixed_pic_rate_flag shall be 1 field_display_target shall be 0	3
9..15	Reserved		

pic_struct vs. display_output_indication SEI

AVC-style				display_output_indication			
Pic_ struct	Output order	Num ClocksTS	ct_type	Field pic flag	Progressive source flag	Field displ pattern	Frame dis. pattern
0	Prog. frame	1	(0)	0	1	-	0
1	Top		0	1	1	0,2	-
			1		0	4	
2	Bot		0		1	1,3	
			1		0	5	
3	Top,bot	2	0	0	1	-	1
			1		0		
4	Bot,top		0		1		2
			1		0		
5	Top,bot,top	3	(0)		1		3
6	Bop,top,bot						4
7	Frame doubling	2					5
8	Frame tripling	3					6

Application model layer

- Always assumes 1080i (540v@60 fields/sec) as output of Level <= 4.1 decoded bitstream
  - Is lossless, regardless of whether each picture was coded as part of a field or frame sequence due to 1080p30 upper bound of level <= 4.1.
  - Progressive frames can be stitched back into 1080p30 with metadata
- Timing (PTS, frame rate, etc.) always assumes ~60 Hz regardless of whether bitstream is field sequence (field_seq_flag=1) or frame sequence (field_seq_flag=1)

Display models

- Progressive display model assumes 1080p60 video output:
  - Copies progressive frames verbatim (marked progressive_source_flag=1). Weaves two field pictures together with POC, POC+/-1 and opposite parity relationship, without loss.
  - De-interlaces each interlaced 540v field (progressive_source_flag=0) into full 1080v frame.
- Interlace model simply passes through Application layer field sequence

Standards elements correspondence

	MPEG-2 | H.262	AVC | H.264	HEVC | H.265
Frame rate signalling	frame_rate	fixed_frame_rate_flag num_units_in_tick	fixed_pic_rate_flag num_units_in_tick
Field from frame output order	top_field_first	pic_struct	display_pattern
Repetition control	repeat_first_field	pic_struct	display_pattern (frame seq.) duplicate_flag (field sequences)
Frame or field structured picture	pict_struct	field_pic_flag	field_seq_flag = field_pic_flag
Progressive / interlace tool switch at seq. layer	progressive_seque nce	frame_mbs_only_flag	N/A
Content type	progressive_frame	ct_type	progressive_source_flag

AVC, MPEG-2, HEVC translation

Output pattern	AVC			MPEG-2					HEVC (proposed)			
Output order	Pic_struct	NumClockTS	ct_type (0=prog, 1=interlace)	Top_field_first	Repeatt first fld	Pict_struct	Prog_frame	Prog_sequence	Field_pic_flag	Prog_src_flag	field_dis_patrn	frame_dis_ptrn
Prog. frame	0	1	(0)	-	-	11	1	0,1	0	1	-	0
Top field (progressive)	1	1	0	-	-	01	1	0	1	1	0,2	-
Top field (interlaced)	1	1	1	-	-	01	0	0	1	0	4	-
Bottom field (progressive)	2	1	0	-	-	10	1	0	1	1	1,3	-
Bottom field (interlaced)	2	1	1	-	-	10	0	0	1	0	5	-
Top, bottom (progressive)	3	2	0	1	0	11	1	0	0	1	-	1
Top, bottom (interlaced)	3	2	1	1	0	11	0	0	0	0	-	1
Bottom, Top (progressive)	4	2	0	0	0	11	1	0	0	1	-	2
Bottom, top (interlaced)	4	2	1	0	0	11	0	0	0	0	-	2
Top, bottom, top	5	3	(0)	1	1	11	1	0	0	1	-	3
Bottom, top, bottom	6	3	(0)	0	1	11	1	0	0	1	-	4
double	7	2	(0)	0	1	11	1	1	0	1	-	5
triple	8	3	(0)	1	1	11	1	1	0	1	-	6

Frame-sequence 1080i recommended settings

	Element	section	Recommended value
level	general_level_idc	Profile_tier_level()	4.0 or 4.1
picture width	pic_width_in_luma_samples	seq_parameter_set (SPS)	1920
picture height	pic_height_in_luma_samples		1080 (assuming minCU is 8)
Picture rate	pic_duration_in_tc_minus1	hrd_paramaters()	~1/60sec (1001 in SCTE)
Sequence structure type	field_seq_flag / field_pic_flag	display_output_indication() SEI	0
Cadence	frame_display_pattern		Field output alternates between top, bottom; also across SAFF
Content type	progressive_source_flag		1 (not recommended to code interlace content in frame seq)

Field-sequence 1080i recommended settings

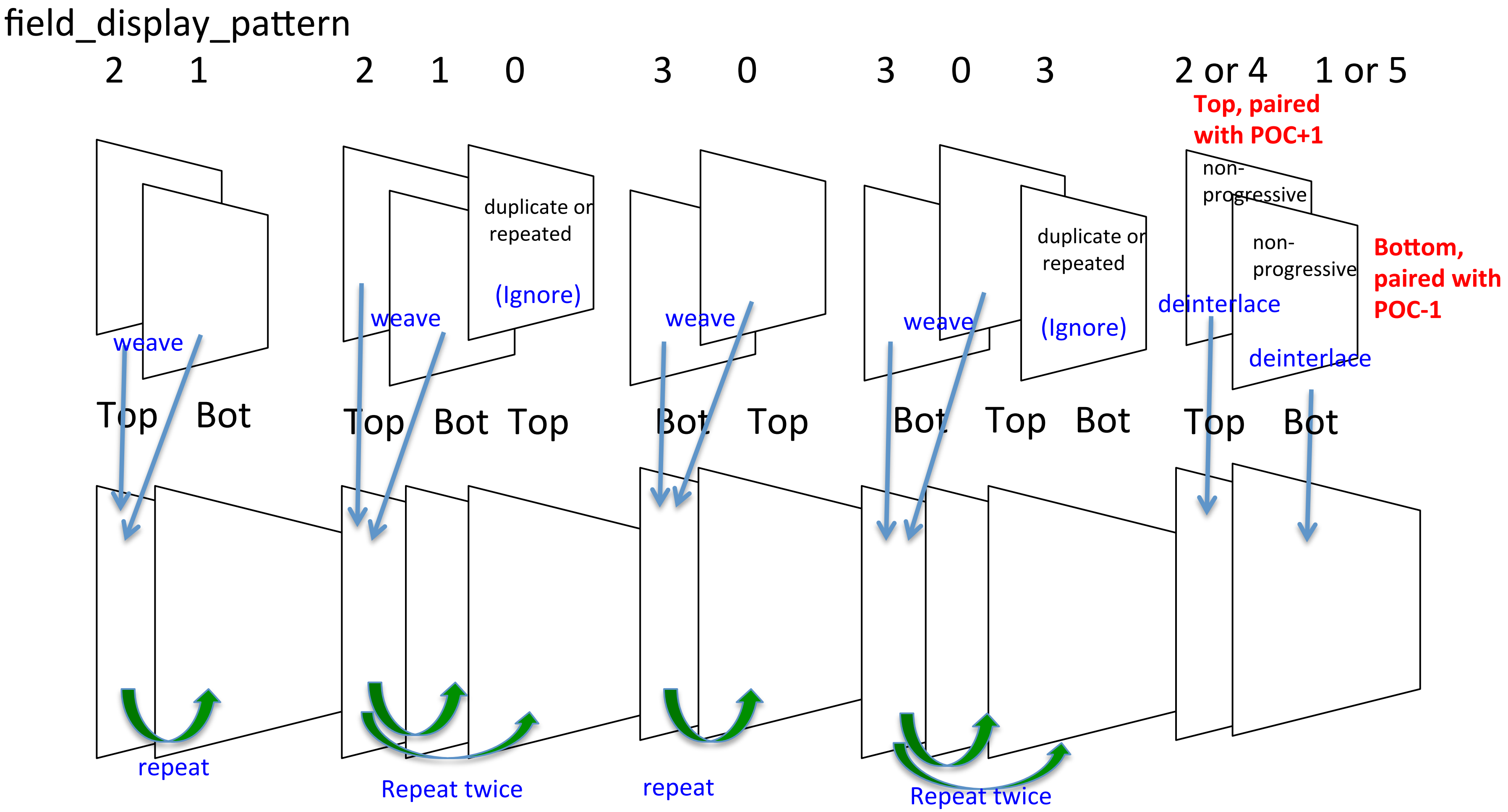
	Element	section	Recommended value
level	general_level_idc	Profile_tier_level()	4.0 or 4.1
picture width	pic_width_in_luma_samples	seq_parameter_set (SPS)	1920
picture height	pic_height_in_luma_samples		544 (assuming minCU is 8)
Picture rate	pic_duration_in_tc_minus1	hrd_paramaters()	~1/60 sec (1001 in SCTE)
Sequence structure type	field_seq_flag / field_pic_flag	display_ouput_indication() SEI	1
Cadence	field_display_pattern		Alternate between top and bottom also across SAFF
Content type	progressive_source_flag		0 for interlaced field 1 for progressive field

When to weave, repeat, deinterlace a <= 30 frames/sec output seq. to 60 frames/sec progreissive display out

- Duplicate_flag=1 fields are dropped, but used as clue to repeat previous display order progressive frame
  - Ideally content of coded duplicate should be checked to see if closely matches previous non-duplicate same-parity field
  - In practice, there should be no coded frames (field_pic_flag=0) marked duplicate (duplicate_flag=1) in <= 30 fps sequences.
- Progressive field-pairs are woven and repeated once (NumClockTS=2) or twice (NumClockTS=3)
- Interlace fields (progressive_source_flag=0) are deinterlaced

60 field/s output to 60 frame/s prog. display

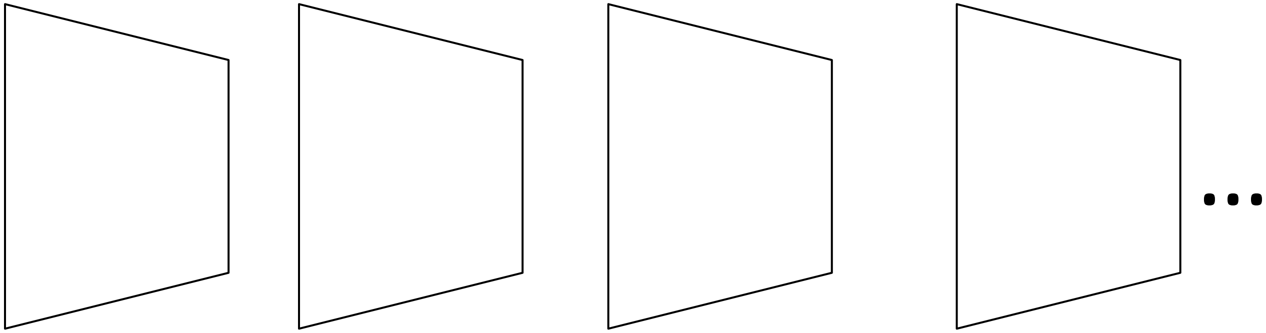
1:1 correspondence between <= level 4.1 output fields and 1080v60frame/sec display output frames



3:2 pulldown 4 film frame →10 output field epoch in a frame seq.

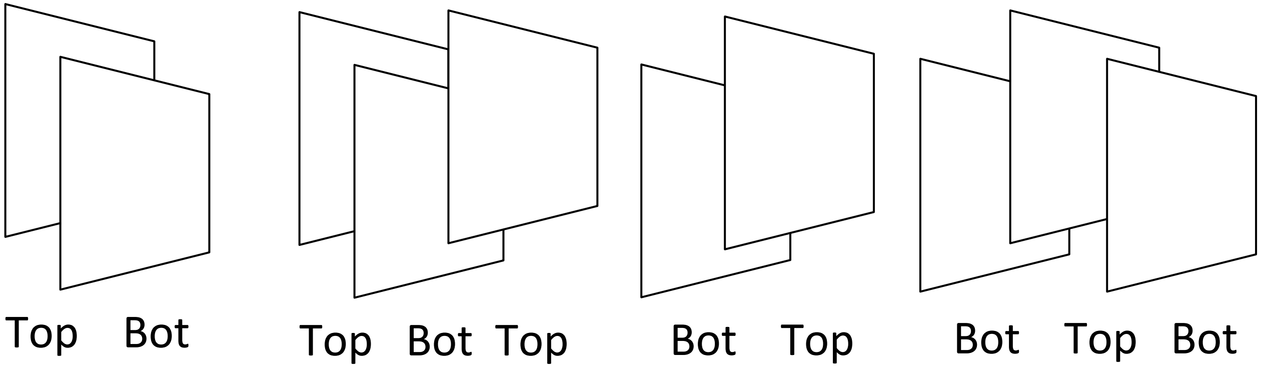
Encoded frame sequence

`field_seq_flag=0`

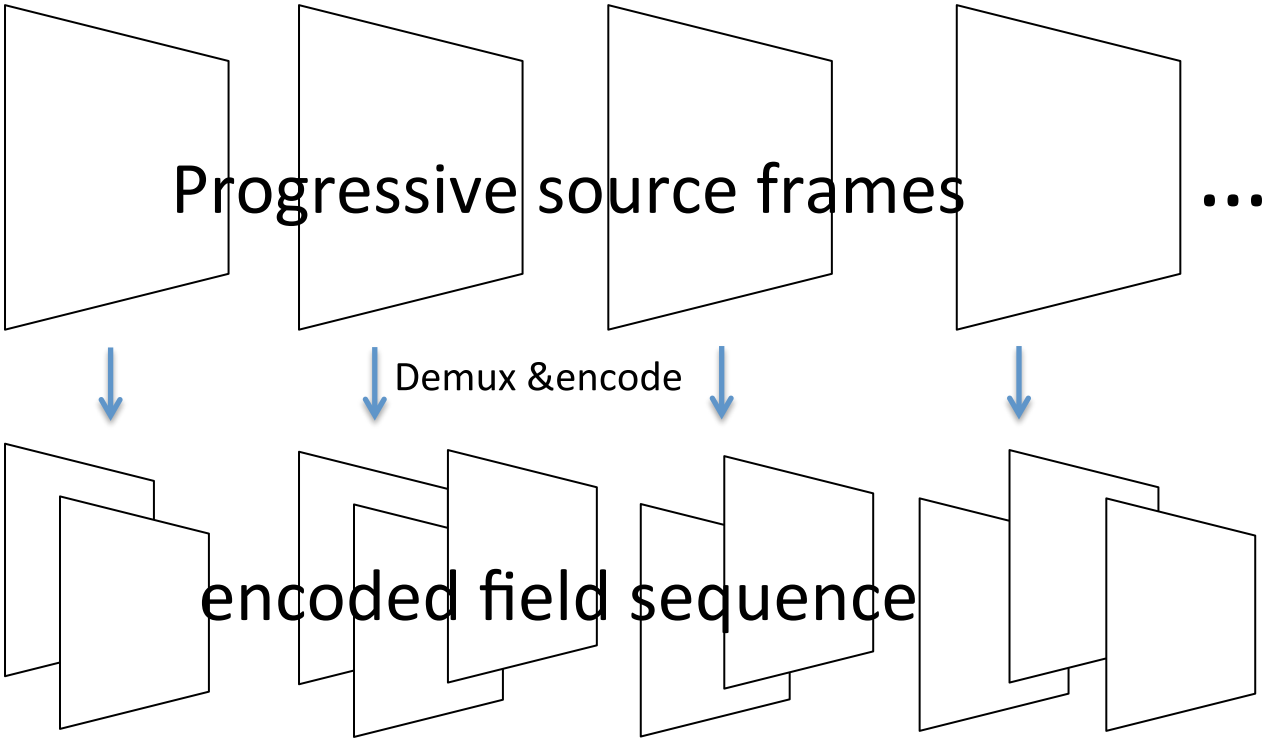


`prog_source_flag`	1	1	1	1
`duplicate_flag`	0	0	0	0
`Frame_displ_patrn`	**1**	**2**	**3**	**4**
`POC (example)`	+0	+2	+5	+7

Padded field display output (app. layer)



3:2 pulldown in a field sequence



Progressive source frames …

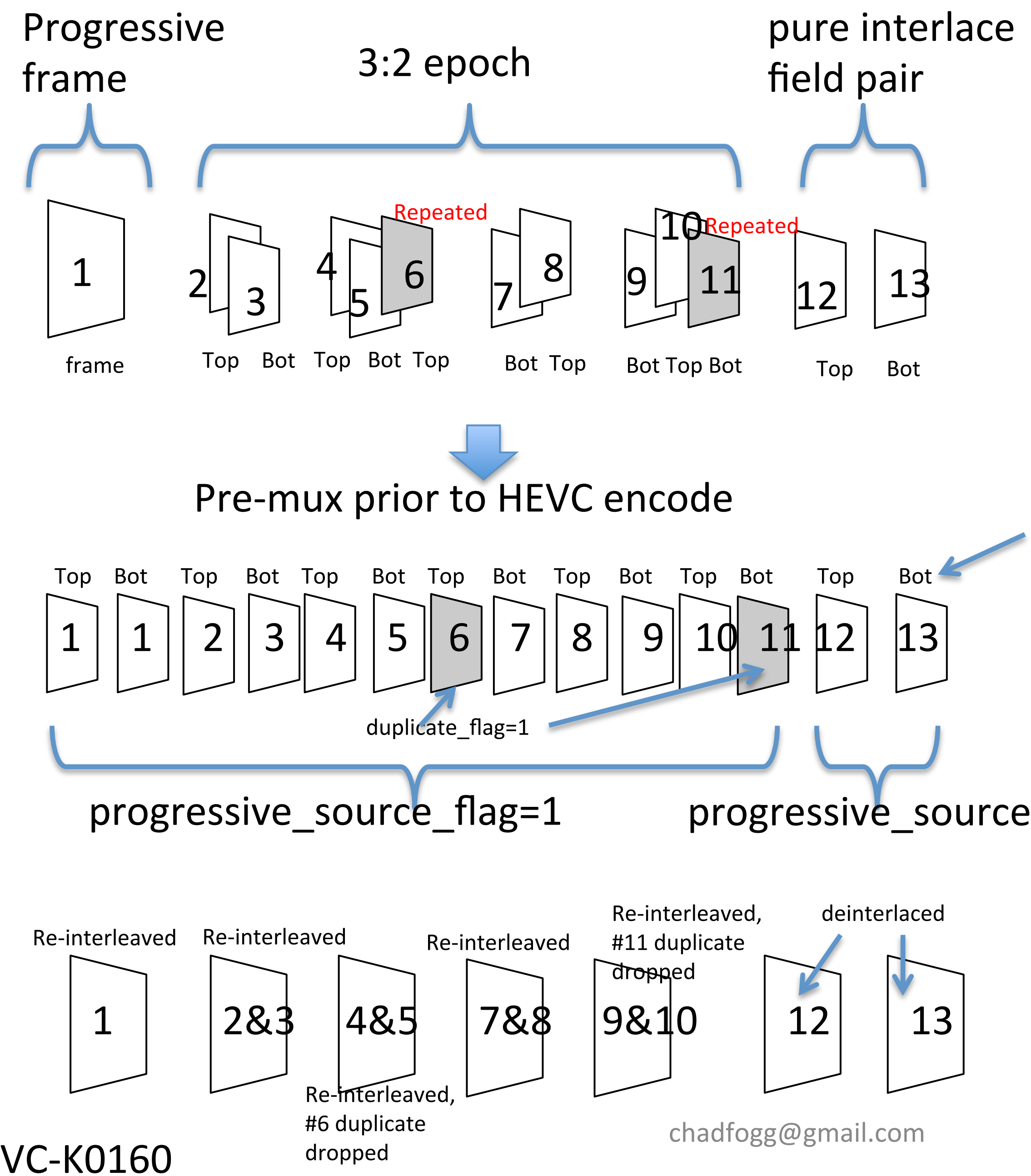
Demux &encode

field_seq_flag=1

encoded field sequence

	Top	Bot	Top	Bot	Top	Bot	Top	Bot	Top	Bot
prog_source_flag	1	1	1	1	1	1	1	1	1	1
field_disp_pattrn	2	1	2	1	0	3	0	3	0	1
duplicate_flag	0	0	0	0	**1**	0	0	0	0	**1**
POC	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9

Frame seq -> field seq → prog display



Progressive frame

3:2 epoch

pure interlace field pair

Pre-mux prior to HEVC encode

duplicate_flag=1

progressive_source_flag=1

progressive_source_flag=0

Re-interleaved

Re-interleaved

Re-interleaved

Re-interleaved, #11 duplicate dropped

deinterlaced

Re-interleaved, #6 duplicate dropped

field_display_pattern

Source

(example: mixed content)

Picture numbers are to show mapping from source to field input to encoder.. It is not POC, frame_num, etc.

Transport & encoding

(all fields in a field sequence)

field_seq_flag=1

Progressive display

Frame sequences for > 30fps progressive displays

Model 3: progressive coding -> prog. display

- Frame doubling & tripling (pic_struct / frame_display_pattern) temporally pad ~24 Hz film content to ~60 Hz display.

Model 4: progressive coding->interlaced display

- Just downscale each 1/60 padded sequence frame to one field.

A separate display_target_type flag in vui_parameters() indicates whether the bitstream is intended for progressive (display_target_type=0) or interlaced display (display_target_type=1)

Issues

1. Though just a model (not normative), HEVC HRD suggests discarding of DPB at SAFF transition.
   - Streams with pic_struct will likely follow a different HRD model anyway
2. DVB, SCTE, etc. statements quoted earlier *encourage* glitches at SAFF transitions.
3. Enforcing no dangling fields in a video coding layer that has no concept of fields or frames.
4. Certain combinations are meaningless (e.g. repeat_pattern = top/bot, etc. when frame rate > 30 Hz)

Terminology

- Coded video sequence
- Padded output sequence
- Display sequence
- Picture
- Field
- Frame
- Field sequence
- Frame sequence