White Paper:

H.264 / AVC Picture Management

Iain Richardson

Vcodex

1 Introduction

This document introduces the parameters and processes involved in managing coded frames within the H.264/AVC standard. This document is informative only and readers should refer to the standard for accurate definitions of the parameters and processes described here.

A frame or field of video is decoded from an access unit (a series of NAL units including one or more coded slices making up a coded picture). The decoding order of access units is indicated by the parameter frame_num (section 2) and the display order is indicated by the parameter Picture Order Count (section 3). Decoded pictures may be marked as "used for reference" (section 4.1) in which case they are available for inter prediction of further decoded pictures. Reference pictures are organised into one or two lists for inter prediction of P, B or SP slices. The default order of these lists (section 4.2) may be explicitly modified by a reference picture list reordering process (section 4.3).

2 frame_num

The parameter frame_num is decoded from each slice header. frame_num increases in **decoding** order of access units and does not necessarily indicate **display** order.

IDR : frame_num set to zero.

Otherwise: increments by 1 from previous reference frame (in decoding order) (unless gaps_in_frame_num_value_allowed, in which case decoder has to create "dummy" decoded frames to fill gap; or unless the current picture and the preceding reference picture are fields with opposite parity).

3 Picture Order Count (POC)

3.1 Overview

POC determines the **display** (output) order of decoded frames, starting from first field of an IDR picture (POC=0).

POC is derived from the slice header in one of 3 ways (see below). POC derived as TopFieldOrderCount and BottomFieldOrderCount, for the top and bottom fields of each coded frame.

Note 1: an application may assign POC proportional to the sampling time of a picture relative to the last IDR. This could lead to variable gaps in POC.

Note 2: the JM reference encoder increments POC by 2 for every complete frame.

3.2 POC Updating

There are 3 supported methods of updating POC:

3.2.1 Type 0: send POC explicitly in each slice header

(Allows maximum flexibility)

TopFieldOrderCount = POCMsb + POCLsb
POCLsb is sent in each slice header
POCMsb is incremented when POCLsb reaches its maximum value

Picture_order_present: delta_POC_bottom is present in slice_header, can change the delta POC between top (first) and bottom (2nd) fields (default is zero delta)

Example (generated from JM reference software)

Frame pictures
Display order: IBPBPBPB… POC type 0 B not used for reference

In order of access units:

Access unit	Type	Used for reference	frame_num	POC_lsb	TopFOC	Display order
1st	I	Yes	0	0	0	0
2nd	P	Yes	1	4	4	2
3rd	B	No	2	2	2	1
4th	P	Yes	2	8	8	4
5th	B	No	3	6	6	3
6th	P	Yes	3	12	12	6
7th	B	No	4	10	10	5
8th	P	Yes	4	16	16	8
…	…					

(Frame number increments relative to previous **reference** picture)

3.2.2 Type 1: set up expected increments in sequence parameter set; only send a delta if there is any change to expected order

(Suitable for situation where there is a repeating "cycle" of pictures, c.f. MPEG-2 GOP)

Sequence parameter set defines number of ref frames in POC "cycle" (repeating group of ref + non-ref frames); offset to each ref frame in the "cycle"; offset to non-reference frame

For each picture, calculate an expected POC as follows:

- calculate number of POC cycles (since last IDR picture)
- calculate position of current frame in POC cycle
- calculate expected POC for current reference frame
- add offset_for_non_ref_pic if this is not a reference frame

TopFieldOrderCount = expected POC + delta_pic_order_cnt[0]
BottomFieldOrderCount = expected POC + delta[1] (if field_pic)
= expected POC + offset to bottom field + delta[0]
(otherwise)

Example (a)

Display order: IBPBPBPB… POC type 1 B not used for reference
1 ref frame in POC cycle; offset to next ref frame = 4; offset for non-ref pic = -2

In order of access units:

Access unit	Type	Used for reference	frame_num	delta_pic_order_cnt[0]	TopFOC	Display order
1st	I	Yes	0	0	0	0
2nd	P	Yes	1	0	4	2
3rd	B	No	2	0	2	1
4th	P	Yes	2	0	8	4
5th	B	No	3	0	6	3
6th	P	Yes	3	0	12	6
7th	B	No	4	0	10	5
8th	P	Yes	4	0	16	8
…	…					

Example (b)

Display order: IBBPBBPBBPB… POC type 1 B not used for reference
1 ref frame in POC cycle; offset to next ref frame = 6; offset for non-ref pic = -4

In order of access units:

Access unit	Type	Used for reference	frame_num	delta_pic_order_cnt[0]	TopFOC	Display order
1st	I	Yes	0	0	0	0
2nd	P	Yes	1	0	6	3
3rd	B	No	2	0	2	1
4th	B	No	2	2	4	2
5th	P	Yes	2	0	12	6
6th	B	No	3	0	8	4
7th	B	No	3	2	10	5
8th	P	Yes	3	0	18	9
…	…					

[Note that 4th and 7th access units have a delta POC of +2; the POC (TopFOC) of each is (expected ref frame count)-4+2]

3.2.3 Type 2: display order same as decoding order

(POC is derived from frame_num; minimal overhead)

For each picture:
if (used for reference)
    set TopFOC and/or BottomFOC to (2*frame_num)
else
    set TopFOC and/or BottomFOC to (2*frame_num)-1

This effectively means that (1) only one non-reference picture can occur between reference pictures, (2) the display order is the same as the access unit order (decoding order), (3) if all pictures are used for reference, POC increments by 2 each time.

4 Reference lists

4.1 Reference picture marking

Picture that is encoded or decoded and available for reference is stored in the Decoded Picture Buffer (DPB) and marked as (a) a short term reference picture, indexed according to frame_num or PicOrderCount or (b) a long term reference picture, indexed according to LongTermPicNum, a reference index assigned when a picture is marked as a long term reference picture. Short term reference pictures may be assigned a LongTermPicNum ("changed" to a long term reference picture) at a later time.

Short term reference pictures are removed from the DPB (a) by an explicit command in the bitstream or (b) when the DPB is "full" (oldest short term picture is removed). Long term pictures are removed by an explicit command in the bitstream.

4.2 Reference picture ordering

Reference pictures are ordered in one or two lists prior to encoding or decoding a slice.

P slices use a single list of reference pictures, list0; B slices use 2 lists, list0 and list1. In each list, short term reference pictures are listed first by default (see below) followed by long term reference pictures (in increasing order of LongTermPicNum). The default short term reference picture order depends on **decoding order** when the current slice is a P slice and depends on **display order** when the current slice is a B slice.

Default order of short term reference pictures in reference lists:

List0 (P slice) : decreasing order of PicNum. (PicNum is a "wrapped around" (mod MaxFrameNum) version of frame_num).

List0 (B slice) : (1) decreasing order of PicOrderCount (for pictures with POC earlier than current picture) then (2) increasing order of PicOrderCount (for pictures with POC later than current picture).

List1 (B slice): (1) increasing order of PicOrderCount (later than current picture) then (2) decreasing order of PicOrderCount (earlier than current picture)

Example:
P slice, list0. Reference picture list is initially empty. Current frame_num is 150. Maximum size of the DPB is 5 frames. Italics indicate a LongTermPicNum.

Operation	list0(0)	list0(1)	list0(2)	list0(3)	list0(4)
Initial state	-	-	-	-	-
Encode frame 150	150	-	-	-	-
Encode 151	151	150	-	-	-
Encode 152	152	151	150	-	-
Encode 153	153	152	151	150	-
Encode 154	154	153	152	151	150
Encode 155	155	154	153	152	151

Assign 154 to LongTermPicNum *3*	155	153	152	151	*3*
Encode 156 and mark it as LongTermPicNum *1*	155	153	152	*1*	*3*
Encode 157	157	155	153	*1*	*3*
…..					

4.3 Reference picture list reordering

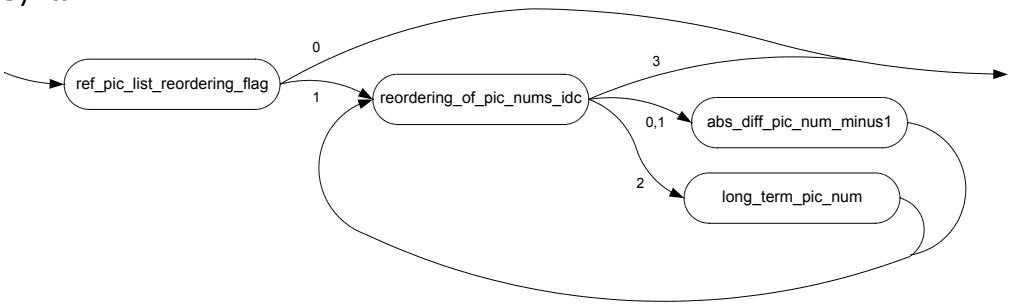
4.3.1 Overview

Purpose: enables encoder to change default order of reference pictures in list0 (and list1 for B-slices) **temporarily** for the next decoded slice.

Example application: The reference picture index ref_idx_l0 (or l1) occurs once in each MB or MB partition. This is signalled as a truncated Exp-Golomb code (te). Larger values of ref_idx cost more bits. There may be a reference picture (short term or long term) that is particularly useful for prediction of the current slice but is not in position 0 in the default list. This process enables the encoder to place this reference picture at a low index in the list so that it costs fewer bits to signal prediction from this picture.

4.3.2 Process overview

Syntax:



If ref_pic_list_reordering_flag is 1, the reordering process is repeatedly carried out until reordering_of_pic_nums_idc is 3.

Reordering process (list0; simplified) (similar for list1):

Initialise a pointer (refIdxL0) to point to the first reference picture index (0)
While reordering_of_pic_nums_idc ≠ 3
    Select a reference picture (short term, indicated by abs_diff_pic_num_minus1, or long term, indicated by long_term_pic_num)
    Move this picture to the position in the list indicated by refIdxL0
    Move all pictures from this position onwards one position later in list
    Increment pointer refIdxL0

4.3.3 Selecting a reference picture to move (remap) to current position

Short-term:
abs_diff_pic_num_minus1 signals an offset (positive or negative) from a **predicted** reference picture. For the first reordering instruction (remapping), the predicted picture is the current picture number. For subsequent reordering instructions, the predicted picture is the picture number of the most recently remapped picture.

If reordering_of_pic_nums_idc is 0, the picture to be remapped is calculated as follows:
remapped_picture = predicted_picture – abs_diff_pic_num_minus_1

If reordering_of_pic_nums_idc is 1, the remapped picture is calculated as follows:
remapped_picture = predicted_picture + abs_diff_pic_num_minus_1

(in each case, the calculation is modified to prevent errors due to the picture number wrapping round).

Long-term:
long_term_pic_num indicates a long term picture to be remapped to the current position in the list.

Example:

P slice, list0, DPB contains 5 reference pictures. Current frame_num is 158 [check: is this correct if latest frame in DPB is 157 ?].
Default reference list is as follows:

157, 155, 153, *1*, *3*

The following series of reference picture reordering commands are received:

Initial predicted_picture = 158; initial refIdxL0 = 0

1. reordering_of_pic_nums_idc = 0, abs_diff_pic_num_minus_1 = 5
remapped_picture = 158 – 5 = 153
New list: 153, 157, 155, *1*, *3*
New predicted picture = 153; new refIdxL0 = 1.

2. reordering_of_pic_nums_idc = 1, abs_diff_pic_num_minus_1 = 2
remapped_picture = 153 + 2 = 155
New list: 153, 155, 157, *1*, *3*
New predicted picture = 155; new refIdxL0 = 2.

3. reordering_of_pic_nums_idc = 2, long_term_pic_num = 3
remapped_picture = *3*
New list: 153, 155, *3*, 157, *1*

4. reordering_of_pic_nums_idc = 3
End of reordering process.

Further reading

Iain E Richardson, "The H.264 Advanced Video Compression Standard", John Wiley & Sons, 2010.

About the author

Iain Richardson wrote the books on H.264 video compression : see http://vcodex.com/h264book/. A founder of OneCodec, he is changing the way video coding works.

Iain Richardson
iain@onecodec.com
http://onecodec.com