|  |  |
| --- | --- |
| **Joint Collaborative Team on 3D Video Coding Extensions**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  6th Meeting: Geneva, CH, 25 Oct. – 1 Nov. 2013 | Document: JCT3V-F0105 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE4-related: ARP reference picture selection and its availability check** | | |
| *Status:* | Input Document to JCT-3V | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Tomohiro Ikai  1-9-2 Nakase, Mihama-ku, Chiba-shi, Chiba 261-8520  JAPAN | Tel: Email: | +81-43-299-8526 [ikai.tomohiro@sharp.co.jp](mailto:ikai.tomohiro@sharp.co.jp) |
| *Source:* | Sharp Corporation | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This contribution proposes to modify derivation method of ARP reference picture and add availability check to guarantees the ARP reference picture exists before ARP prediction process. In this proposal, ARP picture is derived from the first temporal inter picture in each reference picture list (RefPicListX) as opposed to RefPicListX[0]. This modification prevents the problem that if RefPicListX[0] is inter-layer picture (not temporal inter picture), ARP doesn’t work. In terms of availability check, it is proposed to check whether ARP reference picture is in DPB marked as “used for reference”, which is explicated indicated in reference layer’s RPS. It is asserted DPB existence is not sufficient because the DPB removal timing could differ among decoders.

# Introduction

Advanced residual prediction (ARP) utilizes a couple of decoded pictures to estimate the residual. Let me explain the problematic case using the below figure. 

Figure explanation

In the figure, each box indicates a picture and the number in the box indicates the decoding order. When picture 5 is predicted from picture 1, ARP derives residual using the picture 4 (currRefPic) and picture 2 (refIvRefPic).

Problem statement 1

refIvRefPic is chosen based on POC and ViewID and the given POC is that of RefPicList0[0]. However it is possible RefPicList0[0] is inter-layer picture (not temporal inter picture) and ARP doesn’t work. Therefore the ARP reference picture derivation process should be modified to get a temporal inter picture more greedily.

Problem statement 2

Plus, the picture 2 (refIvRefPic) might not be available in the case that picture is not referenced from picture 4(currIvRefPic) and removed from the DPB before picture 5’s decoding. In that case, the ARP should not be invoked. So, available check is needed. Because the DPB removal timing could differ among decoders due to the difference of handling of output timing and implementation, the available check also cares the “used for reference” mark.

Note: the “used for reference ” is marked based on RPS, which is explicitly signalled in all slice header except for IDR and the information is identical among all slices in a picture.

# Proposal

The modification of ARP picture derivation is shown with Yellow highlight.

The availability check is shown with Green highlight.

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize , ctDepth) { | **Descriptor** |
| if( transquant\_bypass\_enable\_flag ) { |  |
| **cu\_transquant\_bypass\_flag** | ae(v) |
| } |  |
| if( slice\_type != I ) |  |
| **skip\_flag**[ x0 ][ y0 ] | ae(v) |
| if( skip\_flag[ x0 ][ y0 ] ) { |  |
| prediction\_unit( x0, y0, log2CbSize ) |  |
| if ( iv\_res\_pred\_flag[ nuh\_layer\_id ] && arpRefPicAvailable ) |  |
| iv\_res\_pred\_weight\_idx | ae(v) |
| if ( icEnableFlag ) |  |
| **ic\_flag** | ae(v) |
| } |  |
| else { |  |
| ,,, |  |
| if( PartMode = = PART\_2Nx2N ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS ) |  |
| if ( iv\_res\_pred\_flag[ nuh\_layer\_id ] && arpRefPicAvailable ) |  |
| **iv\_res\_pred\_weight\_idx** | ae(v) |

The variables arpRefIdxL0, arpRefIdxL1 and arpRefPicAvailable are derived as follows:

arpRefIdxL0 = -1

for(i = 0; i <= num\_ref\_idx\_l0\_active\_minus1 && arpRefIdxL0 < 0; i++)

if (PicOrderCnt(RefPicList0[i]) != PicOrderCntVal)

arpRefIdxL0 = i

arpRefIdxL1 = -1

if (slice\_type != B)

for(i = 0; i <= num\_ref\_idx\_l1\_active\_minus1 && arpRefIdxL1 < 0; i++)

if (PicOrderCnt(RefPicList1[i]) != PicOrderCntVal)

arpRefIdxL1 = i

arpRefPicAvailable = (arpRefPicIdxL0 >= 0) || (arpRefPicIdxL1 >= 0)

**H.8.5.3.3 Decoding process for inter prediction samples**

* + - When predFlagLX is equal to 1, the following applies.
      * The variable refIvRefPicAvailable is derived as follows
        1. If arpRefIdxLX is equal to or larger than 0 and there is a reference picture A in the DPB with PicOrderCnt( picture A ) being equal to PicOrderCntVal, ViewIdx being equal to RefViewIdx[ xP ][ yP ] and marked as “used\_for\_reference”, refIvRefPicAvailable is set equal to 1, Otherwise refIvRefPicAvailable is set equal to 0.
      * The variable resPredFlag is derived as specified in the following:
        1. resPredFlag = ( iv\_res\_pred\_weight\_idx != 0 )   
            ( PicOrderCnt( RefPicListX[ refIdxLX ] ) != PicOrderCntVal ) && refIvRefPicAvailable

(‑168)

* + - * If resPredFlag is equal to 1, the bilinear sample interpolation and residual prediction process as specified in subclause is invoked with the luma locations ( xCb, yCb ), ( xBl, yBl ), the size of the current luma coding block nCbS, the width and the height of the current luma prediction block nPbW, nPbH, the prediction list indication X, the prediction list utilization flag predFlagLX, the reference index refIdxLX, and the motion vectors mvLX, mvCLX, as the inputs and the outputs are the arrays predSamplesLXL, predSamplesLXCb, and predSamplesLXCr.

**H.8.5.3.3.7 Bilinear sample interpolation and residual prediction process**

The process is only invoked if res\_pred\_flag is equal to 1.

Inputs to this process are:

* a luma location ( xCb, yCb ) specifying the top-left sample of the current luma coding block relative to the top left luma sample of the current picture,
* a luma location ( xBl, yBl ) specifying the top-left sample of the current luma prediction block relative to the top-left sample of the current luma coding block,
* a variable nCbS specifying the size of the current luma coding block,
* variables nPbW and nPbH specifying the width and the height, respectively, of the current prediction unit,prediction list utilization flags, predFlagL0 and predFlagL1,
* the prediction list indication X,
* the prediction list utilization flag predFlagLX,
* the reference index refIdxLX,
* the motion vectors mvLX, mvCLX

Output of this process are:

* the (nPbW)x(nPbH) array predSamplesLXL,
* the (nPbW / 2)x(nPbH / 2) arrays predSamplesLXCb and predSamplesLXCr.

The motion vector mvLX is scaled as specified in the following:

* 1. tx = ( 16384 + ( Abs( td ) >> 1 ) ) / td (H‑207)
  2. distScaleFactor = Clip3( −4096, 4095, ( tb \* tx + 32 ) >> 6 ) (H‑208)
  3. mvLX = Clip3( −32768, 32767, Sign( distScaleFactor \* mvLX ) \*   
      ( ( Abs( distScaleFactor \* mvLX ) + 127 ) >> 8 ) ) (H‑209)

where td and tb are derived as:

* 1. td = Clip3( −128, 127, DiffPicOrderCnt( currPic, RefPicListX[ refIdxLX ] ) ) (H‑210)
  2. tb = Clip3( −128, 127, DiffPicOrderCnt( currPic, RefPicListX[ arpRefIdxLX ] ) ) (H‑211)

The location ( xP, yP ) is derived by:

* 1. xP = xCb + xBl (H‑212)
  2. yP = yCb + yBl (H‑213)

The motion vector mvCLX is set equal to mvLX.

The arrays predSamplesLXL, predSamplesLXCb, and predSamplesLXCr  are derived as specified in the following:

* The reference picture consisting of an ordered two-dimensional array refPicLXL of luma samples and two ordered two-dimensional arrays refPicLXCb and refPicLXCr of chroma samples is derived by invoking the process specified in subclause 8.5.2.2.1 with refIdxLX equal to arpRefIdxLX as input.
* The arrays predSamplesLXL, predSamplesLXCb, and predSamplesLXCr are derived by invoking the bilinear sample interpolation process specified in subclause H.8.5.3.3.7.1 with the luma locations ( xCb, yCb ), ( xBl, yBl ), , the luma prediction block width nPbW, the luma prediction block height nPbH,, the motion vectors mvLX, mvCLX, and the reference arrays with refPicLXL, refPicLXCb and refPicLXCr. as the inputs.

The arrays currIvSamplesLXL, currIvSamplesLXCb, and currIvSamplesLXCr  are derived as specified in the following:

* Let currIvRefPic be the picture with PicOrderCnt( currIvRefPic ) equal to PicOrderCntVal and ViewIdx equal to RefViewIdx[ xP ][ yP ].
* Let the reference picture sample arrays currIvRefPicLXL, currIvRefPicLXCb, and currIvRefPicLXCr corresponding to decoded sample arrays SL, SCb, SCr derived in subclause 8.7 for the previously-decoded picture currIvRefPic.
* The arrays currIvSamplesLXL, currIvSamplesLXCb, and currIvSamplesLXCr are derived by invoking the bilinear sample interpolation process specified in subclause H.8.5.3.3.7.1 with the luma locations ( xCb, yCb ) and ( xBl, yBl ), the luma prediction block width nPbW, the luma prediction block height nPbH, the motion vectors mvLX equal to MvDisp[ xP ][ yP ] and mvCLX equal to MvDisp[ xP ][ yP ], and the, and the reference arrays with currIvRefPicLXL, currIvRefPicLXCb and currIvRefPicLXCr as the inputs.

The arrays refIvSamplesLXL, refIvSamplesLXCb, and refIvSamplesLXCr  are derived as specified in the following:

* Let refIvRefPic be the picture with PicOrderCnt( refIvRefPic ) equal to PicOrderCnt( RefPicListX[ arpRefIdxLX ] ) and ViewIdx equal to RefViewIdx[ xP ][ yP ].

# Simulation results

This proposal is implemented in HTM8. The simulation shows that this proposed doesn’t change coding performance.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | video 0 | video 1 | video 2 | video PSNR / video bitrate | video PSNR / total bitrate | synth PSNR / total bitrate | enc time | dec time | ren time |
| Balloons | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 99.9% | 95.5% | 96.3% |
| Kendo | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 98.8% | 93.5% | 94.6% |
| Newspaper\_CC | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 100.4% | 99.7% | 98.6% |
| GT\_Fly | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 99.4% | 98.2% | 98.1% |
| Poznan\_Hall2 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 99.8% | 101.4% | 99.6% |
| Poznan\_Street | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 99.1% | 97.9% | 98.8% |
| Undo\_Dancer | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 99.3% | 100.0% | 99.3% |
| 1024x768 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 99.7% | 96.3% | 96.5% |
| 1920x1088 | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 99.4% | 99.4% | 99.0% |
| **average** | **0.0%** | **0.0%** | **0.0%** | **0.0%** | **0.0%** | **0.0%** | **99.5%** | **98.0%** | **97.9%** |
|  |  |  |  |  |  |  |  |  |  |
| Shark | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 100.4% | 94.0% | 96.2% |

Note decoding time for Shark might be not accurate.

# References

[1] L. Zhang, Y. Chen, X. Li, M. Karczewicz, “CE4: Advanced residual prediction for multiview coding” JCT3V-D0177, JCT3V 4th Meeting: Incheon, KR, 20–26 Apr. 2013

# Conclusion

In this proposal, ARP picture is derived from the first temporal inter picture in each reference picture list (RefPicListX) as opposed to RefPicListX[0]. It is also proposed to check whether ARP reference picture is in DPB marked as “used for reference”.

# Patent rights declaration(s)

**Sharp Corporation may have current or pending patent rights relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**