|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11**  12th Meeting: Geneva, CH, 14–23 Jan. 2013 | Document: JCTVC-L0078 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **TE3: Results of Test 4.6.2.1 on Generalized Residual Prediction** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Xiang Li  Jianle Chen  Krishna Rapaka  Marta Karczewicz | Tel:  Email: | +1-858-658-3923  [lxiang@qti.qualcomm.com](mailto:lxiang@qti.qualcomm.com)  [cjianle@qti.qualcomm.com](mailto:cjianle@qti.qualcomm.com)  [krapaka@qti.qualcomm.com](mailto:krapaka@qti.qualcomm.com)  [martak@qti.qualcomm.com](mailto:martak@qti.qualcomm.com) |
| *Source:* | Qualcomm Incorporated | | |

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

# Abstract

This document reports SHVC TE3 results of test 4.6.2.1 on generalized residual prediction (GRP). GRP is a predictive coding tool which utilizes previously coded base layer and enhancement layer pictures to form the prediction of enhancement layer picture. It is reported that the average luma BD-rate reduction (the average of three RA cases and three LD-P cases) is 2.4%, 2.8%, 3.5%, 3.1%, and 3.4% for five test configurations, respectively.

# Introduction

Generalized residual prediction was proposed in [1][2] to improve the coding efficiency of SHVC. It uses previously coded base layer and enhancement layer pictures to predict the current enhancement layer picture. In the following subsections, the details of GRP are introduced.

## Generalized residual prediction

The framework of GRP is shown in Figure 1 where the case of uni-prediction is illustrated.

Let **B**e and **B**b denote the current block in enhancement layer and its up-sampled collocated base layer block, respectively. Let **P**e0 denote the temporal prediction for the block **B**e obtained by using the motion vector MVe0 (where sub-index 0 refers to reference list 0). Similarly, **P**b0 represents the temporal prediction for the block **B**b obtained by using the same motion vector MVe0 on up-sampled base layer reference picture. Then, the inter predicted residue of the base layer block is obtained as

|  |  |  |
| --- | --- | --- |
|  |  |  |

Considering the temporal prediction **P**e0 for block **B**e, the final uni-prediction **P** for block **B**e is

|  |  |  |
| --- | --- | --- |
|  |  |  |

where *w* is a weighting factor, which takes the values 0, 0.5, or 1.

When extending this method to the bi-predictional case, the final prediction **P** for **B**e is formed as

|  |  |  |
| --- | --- | --- |
|  |  |  |

where sub-index 1 indicates reference list 1.

Generally, weighting factor indicates the prediction mode of GRP. When *w*=0, GRP reduces to single-layer temporal prediction.



**Figure 1 GRP in uni-prediction**

In (2) both block and need interpolation when motion vector points to sub-pixel position. In fact, (2) can be rewritten to (4),

|  |  |  |
| --- | --- | --- |
|  |  |  |

Considering the motion vectors of block and are always the same, the interpolation can be performed directly to differential signal so that only one interpolation is needed and the computational complexity is reduced. Due to the dynamic range requirement of HEVC interpolation module, the values of the differential signal are clipped to have same 8-bit dynamic range as original signal. When , the calculation of (4) is actually conducted as to avoid undesired rounding error. In this case, the values in differential block are clipped to [0, 255]. When , the values in differential block are clipped to [-128, 127].

For bi-predictional case, the same simplification is also applied.

## Signaling of GRP weighting mode

GRP coding mode (weighting factor *w*) is signaled at CU level as a weighting index. The weighting index 0, 1 and 2 are used to indicate the weighting factors 0, 0.5 and 1, respectively. Truncated unary binarization is applied to the weighting mode before CABAC coding. In the case of sub-test 1 wherein only weighting factors 0 and 0.5 are applied, a flag is signaled instead of an index.

## 4-tap up-sampling filter

A short tap up-sampling filter may be applied to obtain up-sampled base layer picture/block for GRP. As a trade-off between coding performance and complexity, 4-tap up-sampling filter shown in Table 1 is employed for luma component of GRP mode in this proposal.

Table 1 4-tap up-sampling filter for GRP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Filter coefficients | | | |
| 0 | 9 | 46 | 9 | 0 |
| 1/3 | 2 | 38 | 25 | -1 |
| 1/2 | 0 | 32 | 32 | 0 |
| 2/3 | -1 | 25 | 38 | 2 |

## Fast GRP mode selection at encoder side

In practice, not all GRP weighting modes need to be checked for each CU. The following fast GRP mode selection method was tested to study the potential encoder acceleration.

Firstly when left, top, and top-right CUs are all larger than the current CU, only one weighting mode is checked.

Secondly for non-merge PUs at temporal level higher than one, two weighting modes are checked at most.

Finally for CUs with non-2Nx2N partition type, two weighting modes are checked at most.

## Additional motion estimation at encoder side

In the default implementation, motion estimation is conducted only when (i.e. normal pixel domain) and the resulting motion information is shared by all GRP weighting modes to keep a relatively low encoding complexity. To achieve better coding performance, motion estimation can be performed for each GRP weighting mode. For , the additional motion estimation is in conducted in related differential pictures.

# Test Results

In this section, GRP is experimentally verified with five different configurations. All tests are based on SHVC common test conditions defined by AHG10. Thanks LGE and ETRI for crosschecking the tests.

## Test 1

In this test, GRP with only two weighting modes, i.e., , is tested. Fast GRP mode selection, 4-tap up-sampling filter, and additional motion estimation are all disabled.

The simulation results are as follows. On average, 1.83% and 3.00% luma BD-rate reduction is achieved for RA cases and LD-P cases, respectively.



## Test 2

In this test, GRP with three weighting modes, i.e., , is applied and fast GRP mode selection is also enabled. The 4-tap up-sampling filter and additional motion estimation are disabled.

The simulation results are as follows. On average, 2.26% and 3.32% luma BD-rate reduction is achieved for RA cases and LD-P cases, respectively. Compared to Test1, additional gain is obtained by introducing weighting mode . It should be noted that, with the fast GRP mode selection method, the encoding and decoding time of Test 2 is even shorter than that in Test 1.



## Test 3

In this test, GRP with three weighting modes, i.e., , is tested. Fast GRP mode selection and 4-tap up-sampling filter are also enabled while the additional motion estimation is disabled.

The simulation results are as follows. On average, 2.67% and 4.24% luma BD-rate reduction is achieved for RA cases and LD-P cases, respectively. Compared to Test 2, additional gain is due to the employment of 4-tap up-sampling filter. Since the up-sampling filter is additionally applied at picture level, the decoding time is increased much.



## Test 4

In this test, GRP with two weighting modes, i.e., , is tested with additional motion estimation enabled. Fast GRP mode selection and 4-tap up-sampling filter both disabled. Please note that there is no decoder side change in this test when compared to Test 1.

The simulation results are as follows. On average, 2.37% and 3.76% luma BD-rate reduction is achieved for RA cases and LD-P cases, respectively. Compared to Test 1, additional gain is due to the additional motion estimation in differential pictures. For both RA and LD-P cases, the average gain is increased by 0.5%-0.7%. However, the encoding time is increased by 30-50%.



## Test 5

In this test, GRP with three weighting modes, i.e., , is tested with additional motion estimation and fast GRP mode selection enabled. The 4-tap upsampling filter is disabled. Please note that there is no decoder side change in this test when compared to Test 2.

The simulation results are as follows. On average, 2.75% and 4.12% luma BD-rate reduction is achieved for RA cases and LD-P cases, respectively. Compared to Test 2, additional gain is due to the additional motion estimation. On average, 0.5% and 0.8% luma BD-rate reduction is obtained over Test 2 for RA and LD-P at the cost of additional 25%-40% encoding time.



# Proposed syntax and semantics

|  |  |
| --- | --- |
| coding\_unit( x0, y0, log2CbSize ) { | Descriptor |
| if( transquant\_bypass\_enable\_flag ) |  |
| **cu\_transquant\_bypass\_flag** | ae(v) |
| if( slice\_type != I ) |  |
| **cu\_skip\_flag**[ x0 ][ y0 ] | ae(v) |
| nCbS = ( 1 << log2CbSize ) |  |
| if( cu\_skip\_flag[ x0 ][ y0 ] ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS ) |  |
| if( LayerID > 0 ) |  |
| **grp\_weight\_idx** | ae(v) |
| } |  |
| else { |  |
| if( slice\_type != I ) |  |
| **pred\_mode\_flag** | ae(v) |
| if( CuPredMode[ x0 ][ y0 ] != MODE\_INTRA | | log2CbSize = = Log2MinCbSizeY ) |  |
| **part\_mode** | ae(v) |
| if( CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA ) { |  |
| if( PartMode = = PART\_2Nx2N && pcm\_enabled\_flag &&  log2CbSize >= Log2MinIpcmCbSizeY &&  log2CbSize <= Log2MaxIpcmCbSizeY ) |  |
| **pcm\_flag**[ x0 ][ y0 ] | ae(v) |
| if( pcm\_flag[ x0 ][ y0 ] ) { |  |
| while( !byte\_aligned( ) ) |  |
| **pcm\_alignment\_zero\_bit** | f(1) |
| pcm\_sample( x0, y0, log2CbSize ) |  |
| } else { |  |
| pbOffset = ( PartMode = = PART\_NxN ) ? ( nCbS / 2 ) : nCbS |  |
| for( j = 0; j < nCbS; j = j + pbOffset ) |  |
| for( i = 0; i < nCbS; i = i + pbOffset ) |  |
| **prev\_intra\_luma\_pred\_flag**[ x0 + i ][ y0+ j ] | ae(v) |
| for( j = 0; j < nCbS; j = j + pbOffset ) |  |
| for( i = 0; i < nCbS; i = i + pbOffset ) |  |
| if( prev\_intra\_luma\_pred\_flag[ x0 + i ][ y0+ j ] ) |  |
| **mpm\_idx**[ x0 + i ][ y0+ j ] | ae(v) |
| else |  |
| **rem\_intra\_luma\_pred\_mode**[ x0 + i ][ y0+ j ] | ae(v) |
| **intra\_chroma\_pred\_mode**[ x0 ][ y0 ] | ae(v) |
| } |  |
| } else { |  |
| if( PartMode = = PART\_2Nx2N ) |  |
| prediction\_unit( x0, y0, nCbS, nCbS ) |  |
| else if( PartMode = = PART\_2NxN ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS / 2 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 2 ), nCbS, nCbS / 2 ) |  |
| } else if( PartMode = = PART\_Nx2N ) { |  |
| prediction\_unit( x0, y0, nCbS / 2, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0, nCbS / 2, nCbS ) |  |
| } else if( PartMode = = PART\_2NxnU ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS / 4 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 4 ), nCbS, nCbS \*3 / 4 ) |  |
| } else if( PartMode = = PART\_2NxnD ) { |  |
| prediction\_unit( x0, y0, nCbS, nCbS \*3 / 4 ) |  |
| prediction\_unit( x0, y0 + ( nCbS \* 3 / 4 ), nCbS, nCbS / 4 ) |  |
| } else if( PartMode = = PART\_nLx2N ) { |  |
| prediction\_unit( x0, y0, nCbS /4, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS / 4 ), y0, nCbS \*3 / 4, nCbS) |  |
| } else if( PartMode = = PART\_nRx2N ) { |  |
| prediction\_unit( x0, y0, nCbS \*3 / 4, nCbS ) |  |
| prediction\_unit( x0 + ( nCbS \* 3 / 4 ), y0, nCbS / 4, nCbS ) |  |
| } else { /\* PART\_NxN \*/ |  |
| prediction\_unit( x0, y0, nCbS / 2, nCbS / 2) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0, nCbS / 2, nCbS / 2 ) |  |
| prediction\_unit( x0, y0 + ( nCbS / 2 ), nCbS / 2, nCbS / 2 ) |  |
| prediction\_unit( x0 + ( nCbS / 2 ), y0 + ( nCbS / 2 ), nCbS / 2, nCbS / 2 ) |  |
| } |  |
| if( LayerID > 0 ) |  |
| **grp\_weight\_idx** | ae(v) |
| } |  |
| if( !pcm\_flag[ x0 ][ y0 ] ) { |  |
| if( CuPredMode[ x0 ][ y0 ] != MODE\_INTRA &&   !(PartMode = = PART\_2Nx2N && merge\_flag[x0][y0]) ) |  |
| **rqt\_root\_cbf** | ae(v) |
| if( rqt\_root\_cbf ) { |  |
| MaxTrafoDepth = ( CuPredMode[ x0 ][ y0 ] = = MODE\_INTRA ?   max\_transform\_hierarchy\_depth\_intra + IntraSplitFlag :   max\_transform\_hierarchy\_depth\_inter ) |  |
| transform\_tree( x0, y0, x0, y0, log2CbSize, 0, 0 ) |  |
| } |  |
| } |  |
| } |  |
| } |  |

**grp\_weight\_idx** specifies weighting mode of generalized residual prediction for a coding unit. The value of grp\_weight\_idx shall be in the range of 0 to 2, inclusive. When not present, grp\_weight\_idx is inferred to be equal to 0.

# Conclusions

In this proposal, the performance of generalized residual prediction for SHVC is investigated in different configurations. By utilizing base layer picture information in predicting enhancement layer picture, significant improvement in coding efficiency is achieved. It is suggested to adopt GRP into SHVC and reference software.

# References

1. J. Chen, K. Rapaka, X. Li, V. Seregin, L. Guo, M. Karczewicz, G. Van der Auwera, J. Sole, X. Wang, C. J. Tu, Y. Chen, “Description of scalable video coding technology proposal by Qualcomm (configuration 1) JCTVC-K0035”, 11th Meeting of Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Shanghai, China, 10-19 Oct., 2012
2. J. Chen, K. Rapaka, X. Li, V. Seregin, L. Guo, M. Karczewicz, G. Van der Auwera, J. Sole, X. Wang, C. J. Tu, Y. Chen, “Description of scalable video coding technology proposal by Qualcomm (configuration 2) JCTVC-K0036”, 11th Meeting of Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, Shanghai, China, 10-19 Oct., 2012.

# Patent rights declaration(s)

**Qualcomm Incorporated 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).**