|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  11th Meeting: Shanghai, China,10 -19 October, 2012 | Document: JCTVC-K0199  M26526 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG7: Sample-based angular intra prediction for HEVC lossless coding** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Minhua Zhou Texas Instruments Inc., USA | Tel: Email:  : | +1-214-480-3816 [zhou@ti.com](mailto:zhou@ti.com) |
| *Source:* | Texas Instruments Inc; | | |

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

# Abstract

Efficient lossless coding is required for real-world applications such as automotive vision, video conferencing and long-distance education. This contribution reports the test results of the sample-based angular intra prediction (SAP) in the HM8.0 environment. The proposed sample-based prediction is exactly same as the HM8.0 block-based angular prediction in terms of prediction angle signaling and sample interpolation, requires no syntax or semantics changes, but differs in decoding process in terms of reference sample selection for interpolation. In the proposed method a sample to be predicted uses its adjacent neighboring samples for better intra prediction accuracy. Compared to the HM8.0 lossless coding method which bypasses transform, quantization, de-blocking filter, SAO and ALF, the proposed method provides an average gain (w/o Class F) of 7.7% in AI-Main, 11.9% in AI-HE10, 1.8% in RA-Main, 2.9% in RA-HE10, 1.5% in LB-Main, 2.3% in LB-HE10, 1.9% in LP-Main and 2.5% in LP-HE10. For class F sequences only, the average gain is 12.1% in AI-Main, 20.7% in AI-HE10, 6.9% in RA-Main, 12.8 % in RA-HE10, 5.5% in LB-Main, 9.6% in LB-HE10, 5.5% in LP-Main and 9.6% in LP-HE10. The SAP is fully parallelized on the encoder side, and can be executed at a speed of one row or one column per cycle on the decoder side.

# Introduction

In HM8.0 the “lossless coding for free” mode is supported in response to increasing needs of lossless video coding for real-world applications, such as automotive vision, video conferencing or long-distance education and content post-production. In HM8.0 the lossless coding is enabled by bypassing transform, quantization and in-loop filters (de-blocking filter, SAO and ALF) and can be turned on/off at CU-level by using lossless coding on/off flag at CU level.

The current HM8.0 lossless coding mode can be improved by introducing new coding tools. One of the tools that has been proposed is so-called sample-based angular intra prediction (SAP) [3][4][5]. Just like the block-based angular intra prediction defined in JCTVC-J1003, in the proposed SAP all the samples in a PU still share a same prediction angle, but the prediction sample of the current sample in a PU uses two of its adjacent neighbors (not limited to the boundary reference samples from neighboring PU). The prediction angle signaling method and the sample interpolation method remain unchanged. Therefore, the SAP introduces changes only in decoding process, but not in syntax and semantics. For the detailed algorithm description of the SAP, please refer to [3][4][5].

This document reports the SAP performance in HM8.0. No technical changes have been introduced when compared to [3][4][5]. In HM8.0 the SAP is integrated in a way in which it can be turned on/off at CU-level, to align with the current HM8.0 lossless coding signaling method.

# Test Settings and Conditions

The simulations of this document have used HM8.0 software, the simulation platform is LSF equipped with Intel(R) Xeon(R) CPU X5570 64 bits Linux machines of different frequencies, the common test conditions and reference configurations specified in [1] are followed, except that only one QP (QP = 0 for Main and QP = -12 for HE10) is tested because QP does not play a role in the lossless coding.

# Experimental results

In Table 1, results of two methods are compared, one is the HM8.0 lossless coding method which bypasses transform, quantization, de-blocking filtering, SAO and ALF, and the other is the proposed method which simply replaces the block-based angular intra prediction with the proposed sample-based intra prediction in lossless coding mode. As shown in Table 1, the proposed method provides an average gain (w/o Class F) of 7.7% in AI-Main, 11.9% in AI-HE10, 1.8% in RA-Main, 2.9% in RA-HE10, 1.5% in LB-Main, 2.3% in LB-HE10, 1.9% in LP-Main and 2.5% in LP-HE10. For class F sequences only, 12.1% in AI-Main, 20.7% in AI-HE10, 6.9% in RA-Main, 12.8 % in RA-HE10, 5.5% in LB-Main, 9.6% in LB-HE10, 5.5% in LP-Main and 9.6% in LP-HE10. (Detailed results can be found in HM-6.0-lossless-Anchor\_vs.\_SAP.xls)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra Main** | | | **All Intra HE10** | | |
|  | **compression ratio** | | Bit-rate saving | **compression ratio** | | Bit-rate saving |
|  | Reference | Tested | Reference | Tested |
| Class A | 2.0 | 2.2 | -8.7% | 1.2 | 1.4 | -12.4% |
| Class B | 2.0 | 2.1 | -5.1% | 1.2 | 1.3 | -8.8% |
| Class C | 1.9 | 2.1 | -7.0% | 1.2 | 1.3 | -11.2% |
| Class D | 1.8 | 2.0 | -8.6% | 1.1 | 1.3 | -12.7% |
| Class E | 2.6 | 3.0 | -10.6% | 1.5 | 1.8 | -16.4% |
| **Overall (w/o F)** | 2.1 | 2.2 | -7.7% | 2.5 | 3.3 | -11.9% |
| **Overall (w/ F)** | 2.5 | 2.7 | -8.5% | 1.2 | 1.4 | -13.4% |
| Class F | 4.5 | 5.2 | -12.1% | 2.5 | 3.3 | -20.7% |
| Enc Time[%] | 109% | | | 116% | | |
| Dec Time[%] | 89% | | | 93% | | |
|  |  |  |  |  |  |  |
|  | **Random Access Main** | | | **Random Access HE10** | | |
|  | **compression ratio** | | Bit-rate saving | **compression ratio** | | Bit-rate saving |
|  | Reference | Tested | Reference | Tested |
| Class A | 2.4 | 2.5 | -2.7% | 1.4 | 1.5 | -3.7% |
| Class B | 2.3 | 2.4 | -1.0% | 1.4 | 1.4 | -2.0% |
| Class C | 2.5 | 2.6 | -1.7% | 1.5 | 1.5 | -3.0% |
| Class D | 2.6 | 2.7 | -2.0% | 1.5 | 1.6 | -3.0% |
| Class E |  |  |  |  |  |  |
| **Overall (w/o F)** | 2.5 | 2.5 | -1.8% | 1.5 | 1.5 | -2.9% |
| **Overall (w/ F)** | 7.4 | 7.9 | -2.8% | 4.2 | 4.8 | -4.8% |
| Class F | 28.5 | 30.7 | -6.9% | 16.0 | 18.8 | -12.8% |
| Enc Time[%] | 104% | | | 105% | | |
| Dec Time[%] | 99% | | | 100% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B Main** | | | **Low delay B HE10** | | |
|  | **compression ratio** | | Bit-rate saving | **compression ratio** | | Bit-rate saving |
|  | Reference | Tested | Reference | Tested |
| Class A |  |  |  |  |  |  |
| Class B | 2.3 | 2.4 | -0.7% | 1.4 | 1.4 | -1.5% |
| Class C | 2.5 | 2.5 | -1.4% | 1.5 | 1.5 | -2.3% |
| Class D | 2.6 | 2.7 | -1.5% | 1.5 | 1.6 | -2.0% |
| Class E | 3.2 | 3.3 | -3.1% | 1.9 | 1.9 | -4.3% |
| **Overall (w/o F)** | 2.6 | 2.7 | -1.5% | 1.5 | 1.6 | -2.3% |
| **Overall (w/ F)** | 11.4 | 11.9 | -2.3% | 6.7 | 7.3 | -3.8% |
| Class F | 46.7 | 49.0 | -5.5% | 27.4 | 29.9 | -9.6% |
| Enc Time[%] | 108% | | | 111% | | |
| Dec Time[%] | 105% | | | 106% | | |
|  |  |  |  |  |  |  |
|  | **Low delay P Main** | | | **Low delay P HE10** | | |
|  | **compression ratio** | | Bit-rate saving | **compression ratio** | | Bit-rate saving |
|  | Reference | Tested | Reference | Tested |
| Class A |  |  |  |  |  |  |
| Class B | 2.3 | 2.3 | -1.1% | 1.4 | 1.4 | -1.8% |
| Class C | 2.4 | 2.5 | -1.5% | 1.5 | 1.5 | -2.4% |
| Class D | 2.6 | 2.6 | -1.6% | 1.5 | 1.6 | -2.0% |
| Class E | 3.1 | 3.2 | -4.0% | 1.9 | 1.9 | -4.2% |
| **Overall (w/o F)** | 2.5 | 2.6 | -1.9% | 1.5 | 1.6 | -2.5% |
| **Overall (w/ F)** | 11.3 | 11.8 | -2.6% | 6.7 | 7.2 | -3.9% |
| Class F | 46.5 | 48.7 | -5.5% | 27.4 | 29.9 | -9.6% |
| Enc Time[%] | 105% | | | 105% | | |
| Dec Time[%] | 101% | | | 98% | | |

Table 1. Simulation results (SAP vs. HM8.0 lossless coding mode)

# Throughput analysis

The sample-based angular intra prediction is fully parallel on the encoder side, and can be executed at a speed of one row or one column per cycle on the decoder side. For a 4x4 intra PU, this would mean 4 cycles per 4x4 block on the decoder side, which is far less than a typical cycle budget of 12 per 4x4 block.

# Conclusion and recommendation

Efficient HEVC lossless coding is required for real-world applications such as automotive vision and video conferencing. The proposed lossless coding method with sample-based angular intra prediction provides significant gain over the HM8.0 lossless coding method which bypasses transform, quantization, de-blocking filter, SAO and ALF. It is recommended to adopt this method into test model for potential specification of HEVC lossless coding profile in the future.

# References

[1] F. Bossen, “Common test conditions and software reference configurations,” JCT-VC Document, JCTVC-J1100, Stockholm, Sweden, July 2012.

[2] [B. Bross](mailto:benjamin.bross@hhi.fraunhofer.de), [W.-J. Han](mailto:wjhan.han@samsung.com), [J.-R. Ohm](mailto:ohm@ient.rwth-aachen.de), [G. J. Sullivan](mailto:garysull@microsoft.com), [T. Wiegand](mailto:thomas.wiegand@hhi.fraunhofer.de) “High Efficiency Video Coding (HEVC) Test Model 8 (HM 8) Encoder Description” JCT-VC Document, JCTVC-J1003, Stockholm, Sweden, July 2012.

[3] M. Zhou, “AHG22: Sample-based angular prediction (SAP) for HEVC lossless coding**”,** JCT-VC Document, JCTVC-G093, Geneva, CH, Nov. 2011

[4] M. Zhou, “AHG19: Method of frame-based lossless coding mode for HEVC**”,** JCT-VC Document, JCTVC-H083, San Jose, CA, USA, February 2012.

[5] M. Zhou, “AHG13: Sample-based angular intra prediction for HEVC lossless coding”, JCT-VC Document, JCTVC-I0117, Geneva, CH, April 2012

# Patent rights declaration(s)

**Texas Instruments, Inc. may have IPR 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).**

# WD text changes

Replace 8.4.4.2.6

##### Specification of Intra\_Angular (2..34) prediction mode

With (changes are marked in yellow)

Inputs to this process are:

– intra prediction mode intraPredMode,

– neighbouring samples p[ x ][ y ], with x, y = −1..2\*nT−1,

– a variable nT specifying the transform block size,

– a variable cIdx specifying the chroma component of the current block,

* the residual sample array values r[ x ][ y ], with x, y = 0..nT−1.

Output of this process are:

– predicted samples predSamples[ x ][ y ], with x, y =0..nT−1.

illustrates the total 34 intra angles and specifies the mapping table between intraPredMode and the angle parameter intraPredAngle.



Figure 8‑2 – Intra prediction angle definition (informative)

Table 8‑4 – Specification of intraPredAngle

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **intraPredMode** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** |
| **intraPredAngle** | - | 32 | 26 | 21 | 17 | 13 | 9 | 5 | 2 | 0 | −2 | −5 | −9 | −13 | −17 | −21 | −26 |
| **intraPredMode** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** | **27** | **28** | **29** | **30** | **31** | **32** | **33** | **34** |
| **intraPredAngle** | −32 | −26 | −21 | −17 | −13 | −9 | −5 | −2 | 0 | 2 | 5 | 9 | 13 | 17 | 21 | 26 | 32 |

further specifies the mapping table between intraPredMode and the inverse angle parameter invAngle.

Table 8‑5 – Specification of invAngle

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **intraPredMode** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** |
| **invAngle** | −4096 | −1638 | −910 | −630 | −482 | −390 | −315 | −256 |
| **intraPredMode** | **19** | **20** | **21** | **22** | **23** | **24** | **25** | **26** |
| **invAngle** | −315 | −390 | −482 | −630 | −910 | −1638 | −4096 | - |

If cu\_transquant\_bypass\_flag is equal to 0, the values of the prediction samples predSamples[ x ][ y ], with x, y = 0..nT−1 are derived as follows.

– If intraPredMode is equal or greater than 18, the following ordered steps apply.

1. The reference sample array ref[ x ], with x= −nT..2\*nT is specified as follows.

ref[ x ] = p[ −1+x ][ −1 ], with x=0..nT (8‑40)

* If intraPredAngle is less than 0, the main reference sample array is extented as follows.
* When ( nT\*intraPredAngle ) >>5 is less than −1,

ref[ x ] = p[ −1 ][ −1+( ( x\*invAngle+128 )>>8 ) ], with x=( nT\*intraPredAngle ) >>5..−1 (8‑41)

* Otherwise,

ref[ x ] = p[ −1+x ][ −1 ], with x=nT+1..2\*nT (8‑42)

1. The values of the prediction samples predSamples[ x ][ y ], with x, y = 0..nT−1 are derived as follows.
   1. The index variable iIdx and the multiplication factor iFact are derived by

iIdx = ( ( y + 1 )\*intraPredAngle ) >> 5 (8‑43)

iFact = ( ( y + 1 )\*intraPredAngle ) && 31 (8‑44)

* 1. Depending on the value of iFact, the following applies.
* If iFact is not equal to 0, the value of the prediction samples predSamples[ x ][ y ] is derived by

predSamples[ x ][ y ] = ( ( 32 – iFact )\*ref[ x+iIdx+1 ] + iFact\*ref[ x+iIdx+2] + 16 ) >> 5 (8‑45)

* Otherwise, the value of the prediction samples predSamples[ x ][ y ] is derived by

predSamples[ x ][ y ] = ref[ x+iIdx+1 ] (8‑46)

* 1. When intraPartMode is equal to Intra\_Angular (26) and cIdx is equal to 0, the following filtering applies with x = 0, y = 0..nT−1.

predSamples[ x ][ y ] = Clip1Y( p[ x ][ −1 ] + ( ( p[ −1 ][ y ] – p[ −1 ][ −1 ] ) >> 1 ) ) (8‑47)

– Otherwise (intraPredMode is less than 18), the following ordered steps apply.

1. The reference sample array ref[ x ], with x= −nT..2\*nT is specified as follows.

ref[ x ] = p[ −1 ][ −1+x ], with x=0..nT (8‑48)

* If intraPredAngle is less than 0, the main reference sample array is extented as follows.
* When ( nT\*intraPredAngle ) >>5 is less than −1,

ref[ x ] = p[ −1+( ( x\*invAngle+128 )>>8 ) ][ −1 ], with x=( nT\*intraPredAngle ) >>5..−1 (8‑49)

* Otherwise,

ref[ x ] = p[ −1 ][ −1+x ], with x=nT+1..2\*nT (8‑50)

1. The values of the prediction samples predSamples[ x ][ y ], with x, y = 0..nT−1 are derived as follows.
2. The index variable iIdx and the multiplication factor iFact are derived by

iIdx = ( ( x + 1 )\*intraPredAngle ) >> 5 (8‑51)

iFact = ( ( x + 1 )\*intraPredAngle ) && 31 (8‑52)

1. Depending on the value of iFact, the following applies.

* If iFact is not equal to 0, the value of the prediction samples predSamples[ x ][ y ] is derived by

predSamples[ x ][ y ] = ( ( 32 – iFact )\*ref[ y+iIdx+1 ] + iFact\*ref[ y+iIdx+2] + 16 ) >> 5 (8‑53)

* Otherwise, the value of the prediction samples predSamples[ x ][ y ] is derived by

predSamples[ x ][ y ] = ref[ y+iIdx+1 ] (8‑54)

1. When intraPartMode is equal to Intra\_Angular (10) and cIdx is equal to 0, the following filtering applies with x = 0..nT−1, y = 0.

predSamples[ x ][ y ] = Clip1Y( p[ −1 ][ y ] + ( ( p[ x ][ −1 ] – p[ −1 ][ −1 ] ) >> 1 ) (8‑55)

If cu\_transquant\_bypass\_flag is equal to 0, the values of the prediction samples predSamples[ x ][ y ], with x, y = 0..nT−1 are derived as follows.

* If intraPredMode is less than 18,
* if intraPredAngle is less than 0, ref [0] = p[-1 + x][ y-1], ref[1] = p[-1+x][ y];
* if intraPredAngle is larger than or equal to 0, ref[0] = p[-1 + x][ y]; if y is less than nT -1, ref[1] = p[-1+x][ y+1], otherwise, ref[1] = p[-1+x][ nT-1];
* Otherwise if intraPredMode is larger than or equal to 18
* if intraPredAngle less than 0, ref[0] = p[-1 + x][ y-1], ref[1] = p[x][ y-1];
* if intraPredAngle large than or equal to 0, ref[0] = p[x][ y-1]; if x is less than nT -1, ref[1] = p[1+x][ y-1], otherwise, ref[1] = p[nT-1, y-1].

– The multiplication factor iFact are derived by

iFact = ( intraPredAngle ) & 31 (8‑57)

– Depending on the value of iFact, the following applies.

* If iFact is not equal to 0, the value of the prediction samples predSamples[ x][y ] is derived by

predSamples[ x][ y ] = ( ( 32 – iFact )\*ref[ 0 ] + iFact\*ref[ 1] + 16 ) >> 5 (8‑58)

* Otherwise, the value of the prediction samples predSamples[ x][ y ] is derived by

predSamples[ x][ y ] = ref[0] (8‑59)

* The sample p[x][ y] is reconstructed by
* p[x][y] = r [x][ y] + predSamples[x][ y]