|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  8th Meeting: San Jose, CA, USA, 1 - 10 February, 2012 | Document: JCTVC-H0083  M22955 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **AHG19: Method of frame-based lossless coding mode for HEVC** | | |
| *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 proposes to install a high-level flag “**bypass\_coding\_mode\_enabled\_flag”** in PPS to enable lossless coding mode by bypassing inverse quantization, inverse transform, de-blocking filter, SAO and ALF, and to use sample-based angular intra prediction (SAP) in lossless coding mode for better coding efficiency. The proposed sample-based prediction is exactly same as the HM5.0 block-based angular prediction in terms of prediction angles and sample interpolation, requires no syntax or semantics changes, but differs in decoding process in terms of reference sample selection. In the proposed method a sample to be predicted uses its direct neighboring samples for better intra prediction accuracy. Compared to the HM5.0 anchor lossless method which bypasses transform, quantization, de-blocking filter, SAO and ALF, the proposed method provides an average gain of 8.30% in AI-HE, 8.53% in AI-LC, 2.85% in RA-HE, 2.93% in RA-LC, 2.0% in LB-HE and 2.06% in LB-LC. For class F sequences only, the average gain is 12.23% in AI-HE, 12.55% in AI-LC, 7.11% in RA-HE, 7.28% in RA-LC, 5.72% in LB-HE and 5.84% in LB-LC. 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

There are increasing needs of lossless video coding for real-world applications. For example, in the automotive vision application, video captured from cameras of a vehicle may need to be transmitted to the center processors losslessly for video analytics purpose. Another example is video conferencing or long-distance education in which hybrid nature and syntactic video coding might be required, where part of video scene might contain syntactic contents such as presentation slides that need to be coded losslessly. In those application scenarios, lossless coding mode which provides certain level of compression is required, and is desirable to enable it in profiles of 4:2:0 chroma format as well. The default lossless coding method is to bypass both transform and quantization on encoder and decoder side. In this contribution, a high-level flag “**bypass\_coding\_mode\_enabled\_flag** “ in PPS and a sample-based angular intra prediction are proposed to provide more efficient coding of lossless coding mode at frame-level.

# Algorithm description

The simple lossless coding mode is to bypass quantization and inverse quantization. Figure 1 illustrates the HEVC encoder diagram with quantization and inverse quantization bypassed. In the lossless mode, deblocking filter, SAO and ALF are also disabled. This lossless mode serves as lossless anchor method in this contribution.

bypass

bypass

-

+

DCT

Q

Entropy coding

IQ

IDCT

De-blocking

SAO

ALF

MC

IP

Frame

buffer

IPE

ME

**LCU**

bypass

bypass

bypass

Figure 1. Diagram of HEVC encoder with losseless coding mode that bypasses transform and quantization, and disables deblocking, SAO and ALF

In HM5.0 a block-based angular intra prediction is defined to explore spatial sample redundancy in intra-coded frame. As shown in Figure 2 a total of 33 angles are defined for the angular prediction. Those angles can be categorized into two classes: vertical and horizontal angular predictions as depicted in Figure 2.



Horizontal angular predictions

Vertical angular predictions

Figure 2. Intra prediction angle definition in HM5.0

For an N x N PU (luma or chorma component), the block-based angular intra prediction involves in a total of 4N+1 reference samples (i.e. brown samples in Figure 3) from neighboring PU to form the prediction block of the current PU. The angular prediction angle is signaled in the bitstream so that decoder can perform exactly the same operations to reconstruct the prediction block on the decoder side.

N

N

NxN PU

Reference samples

Figure 3. Block-based angular intra prediction in HM5.0

For lossless coding, the reference samples are known not only around upper and left boundaries of the current PU, but also within the current PU. Therefore, it is logical to extend the intra angular prediction to sample-level to better explore spatial sample redundancy in lossless coding environment.

In the proposed sample-based intra angular prediction algorithm, all the samples in a PU shares a same prediction angle as defined in HM5.0. Also, the signaling of prediction angles is exactly same as HM5.0. The major deference is that the angular prediction is performed sample by sample for a PU in the proposed method to achieve better intra prediction accuracy. That is, the prediction block for the current PU is generated by performing the sample-based angular prediction sample by sample by using a same prediction angle.

N

N

NxN PU

N

N

NxN PU

1. raster-scanning processing order for vertical sample-based angular predictions
2. vertical-scanning processing order for horizontal sample-based angular predictions

Reference samples

Padded samples

Figure 4. Processing order of sample-based angular intra prediction

In the proposed method, samples in a PU are processed in pre-defined orders so that the neighboring samples are available when the current sample in the PU is being predicted from its direct neighbors, especially on the decoder side. As shown in Figure 4, the raster-scanning and vertical scanning processing order is applied to the vertical and horizontal angular predictions, respectively. The processing of reference samples around the upper and left PU boundaries of the current PU is exactly same as defined in HM5.0, while reference samples around right and bottom PU boundaries of the current PU are simply padded from the closest boundary samples of the current PU (see padded samples in Figure 4).

N

N

NxN PU

N

N

NxN PU

1. Vertical sample-based angular predictions with negative angles
2. Horizontal sample-based angular predictions with negative angles

Reference samples

Padded samples

Figure 5. Reference sample locations relative to the current sample for sample-based angular intra prediction with negative angles

N

N

NxN PU

N

N

NxN PU

(a) Vertical sample-based angular predictions with positive angles

1. Horizontal sample-based angular predictions with positive angles

Reference samples

Padded samples

Figure 6. Reference sample locations relative to the current sample for sample-based angular intra prediction with positive angles

Based on prediction angles defined in Figure 2 (which are exactly same as those defined in HM5.0), at most two reference samples are selected for each sample to be predicted in the current PU. Figure 5 and Figure 6 depict the reference sample locations (i.e. **a** and **b**) relative to the current sample (i.e. **x** to be predicted) for horizontal and vertical sample-based angular prediction with negative and positive predication angles, respectively. Note that depending on the current sample location and prediction angle selected, the reference sample **a** and **b** can be those from neighboring PUs (i.e. brown ones in Figure 5 and 6), padded samples or samples inside the current PU.

Prediction angle

32-iFact

iFact

Figure 7. Bilinear interpolation of sample-based intra angular prediction

Once the reference samples are determined based on prediction angle and current sample location, the actual interpolation for prediction sample generation is defined exactly same as HM5.0. As shown in Figure 7, let a, b be reference samples selected for the current sample x, and iFact be distance from reference sample b to prediction location p (based on the prediction angle selected), the prediction value p for the current sample x is defined as

***p = ((32 – iFact)\*a + iFact \* b + 16)>>5***

Once the prediction sample value p for the current sample x is computed based on the method described above, different operation is carried out on the encoder and decoder sides: on the encoder side residual sample value x – p is generated for the current sample; on the decoder side, the current sample x is reconstructed by adding the decoded residual to prediction sample p, the reconstructed sample x then serves as a reference sample for the angular prediction of rest samples of the current PU.

# Test Settings and Conditions

The simulations of this document have used HM5.0 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 = 22) is tested because QP does not play a role in lossless coding. Although QP is meaningless for lossless coding, it still plays a role on the encoder side due to RDO. In the simulations, RDOQ, deblocking filter, SAO and ALF are also disabled.

# Experimental results

In Table 1, results of two methods are compared, one is the HM5.0 lossless anchor 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 of 8.30% in AI-HE, 8.53% in AI-LC, 2.85% in RA-HE, 2.93% in RA-LC, 2.0% in LB-HE and 2.06% in LB-LC. For class F sequences only, the average gain is 12.23% in AI-HE, 12.55% in AI-LC, 7.11% in RA-HE, 7.28% in RA-LC, 5.72% in LB-HE and 5.84% in LB-LC. (Detailed results can be found in HM-5.0-lossless-anchor\_vs.\_SAP.xls)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | **compression ratio** | | **bit-rate saving** | **compression ratio** | | **bit-rate saving** |
|  | anchor | proposed | anchor | proposed |
| Class A | 2.21 | 2.46 | -10.26% | 2.20 | 2.46 | -10.48% |
| Class B | 2.09 | 2.18 | -4.52% | 2.08 | 2.18 | -4.69% |
| Class C | 1.97 | 2.11 | -6.49% | 1.97 | 2.11 | -6.76% |
| Class D | 1.83 | 2.01 | -8.41% | 1.82 | 2.01 | -8.67% |
| Class E | 2.90 | 3.23 | -10.30% | 2.89 | 3.23 | -10.47% |
| Class F | 4.51 | 5.21 | -12.23% | 4.48 | 5.20 | -12.55% |
| **Overall** | 2.58 | 2.86 | -8.30% | 2.57 | 2.85 | -8.53% |
|  |  | | |  | | |
| Enc Time[%] | 105% | | | 105% | | |
| Dec Time[%] | 93% | | | 93% | | |
|  | | | | | | |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | **compression ratio** | | **bit-rate saving** | **compression ratio** | | **bit-rate saving** |
|  | anchor | proposed | anchor | proposed |
| Class A | 2.72 | 2.80 | -2.99% | 2.72 | 2.80 | -3.07% |
| Class B | 2.38 | 2.41 | -0.94% | 2.38 | 2.41 | -0.99% |
| Class C | 2.55 | 2.59 | -1.69% | 2.55 | 2.59 | -1.76% |
| Class D | 2.67 | 2.73 | -2.07% | 2.66 | 2.72 | -2.13% |
| Class E |  |  |  |  |  |  |
| Class F | 28.38 | 30.90 | -7.11% | 28.24 | 30.82 | -7.28% |
| **Overall** | 7.99 | 8.55 | -2.85% | 7.95 | 8.54 | -2.93% |
|  |  | | |  | | |
| Enc Time[%] | 102% | | | 101% | | |
| Dec Time[%] | 100% | | | 99% | | |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | **compression ratio** | | **bit-rate saving** | **compression ratio** | | **bit-rate saving** |
|  | anchor | proposed | anchor | proposed |
| Class A |  | | |  | | |
| Class B | 2.38 | 2.40 | -0.77% | 2.38 | 2.40 | -0.80% |
| Class C | 2.55 | 2.58 | -1.45% | 2.54 | 2.58 | -1.51% |
| Class D | 2.66 | 2.71 | -1.66% | 2.66 | 2.71 | -1.72% |
| Class E | 4.15 | 4.16 | -0.28% | 4.15 | 4.16 | -0.30% |
| Class F | 47.02 | 49.50 | -5.72% | 46.82 | 49.36 | -5.84% |
| **Overall** | 11.66 | 12.18 | -2.00% | 11.62 | 12.15 | -2.06% |
|  |  | | |  | | |
| Enc Time[%] | 103% | | | 102% | | |
| Dec Time[%] | 100% | | | 100% | | |
|  |  |  |  |  |  |  |

Table 1. Simulation results (proposed method vs. HM5.0 lossless anchor method)

As additional information, performance of HM5.0 lossy coding with QP = 0 is also investigated. As shown in Table 2 below, QP = 0 lossy coding consumes significantly higher bit-rates when compared to the HM5.0 lossless method. (Detailed results are provided in HM-5.0-lossless-anchor\_vs.\_QP0.xls). It is worthy to mention that in RA-HE10 lossycoding at around 60dB provides significantly higher compression efficiency than lossless coding for 10-bit video **Nebuta** and **SteamLocomotive**.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | **compression ratio** | | **bit-rate saving** | **compression ratio** | | **bit-rate saving** |
|  | anchor | proposed | anchor | proposed |
| Class A | 2.21 | 2.06 | 7.20% | 2.20 | 2.07 | 6.56% |
| Class B | 2.09 | 1.87 | 11.79% | 2.08 | 1.87 | 11.40% |
| Class C | 1.97 | 1.74 | 13.15% | 1.97 | 1.74 | 12.80% |
| Class D | 1.83 | 1.66 | 10.64% | 1.82 | 1.66 | 10.35% |
| Class E | 2.90 | 2.55 | 13.54% | 2.89 | 2.58 | 12.05% |
| Class F | 4.51 | 3.78 | 22.91% | 4.48 | 3.76 | 23.09% |
| **Overall** | 2.58 | 2.27 | 13.67% | 2.57 | 2.27 | 13.24% |
|  |  | | |  | | |
| Enc Time[%] | 254% | | | 143% | | |
| Dec Time[%] | 135% | | | 139% | | |
|  | | | | | | |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | **compression ratio** | | **bit-rate saving** | **compression ratio** | | **bit-rate saving** |
|  | anchor | proposed | anchor | proposed |
| Class A | 2.72 | 2.39 | 13.78% | 2.72 | 2.40 | 13.12% |
| Class B | 2.38 | 2.05 | 16.24% | 2.38 | 2.03 | 17.38% |
| Class C | 2.55 | 2.16 | 17.99% | 2.55 | 2.16 | 17.97% |
| Class D | 2.67 | 2.26 | 17.92% | 2.66 | 2.29 | 16.74% |
| Class E |  |  |  |  |  |  |
| Class F | 28.38 | 23.13 | 18.88% | 28.24 | 23.61 | 16.83% |
| **Overall** | 7.99 | 6.59 | 17.26% | 7.95 | 6.69 | 16.81% |
|  |  | | |  | | |
| Enc Time[%] | 138% | | | 102% | | |
| Dec Time[%] | 119% | | | 117% | | |
|  | | | | | | |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | **compression ratio** | | **bit-rate saving** | **compression ratio** | | **bit-rate saving** |
|  | anchor | proposed | anchor | proposed |
| Class A |  | | |  | | |
| Class B | 2.38 | 2.04 | 16.82% | 2.38 | 2.02 | 18.19% |
| Class C | 2.55 | 2.15 | 18.56% | 2.54 | 2.14 | 18.49% |
| Class D | 2.66 | 2.24 | 18.92% | 2.66 | 2.27 | 17.32% |
| Class E | 4.15 | 3.19 | 30.00% | 4.15 | 3.26 | 27.15% |
| Class F | 47.02 | 41.36 | 15.22% | 46.82 | 43.38 | 12.14% |
| **Overall** | 11.66 | 10.14 | 19.24% | 11.62 | 10.55 | 18.21% |
|  |  | | |  | | |
| Enc Time[%] | 150% | | | 119% | | |
| Dec Time[%] | 136% | | | 145% | | |

Table 2. Simulation results (HM5.0 lossy coding w/ QP = 0 vs. HM5.0 lossless anchor method)

# 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 HM5.0 anchor lossless method which bypasses transform, quantization, de-blocking filter, SAO and ALF. It is recommended to adopt this method into test model to provide functionality of efficient lossless coding mode at frame-level.

# References

[1] F. Bossen, “Common test conditions and software reference configurations,” JCT-VC Document, JCTVC-G1200, Geneva, CH, Nov. 2011.

[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) text specification Working Draft 5” JCT-VC Document, JCTVC-G1103, Geneva, CH, Nov. 2011

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

# 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

**In 7.3.2.2** **Picture parameter set RBSP syntax**

Add a flag **bypss\_quant\_transform\_enabled\_flag**

|  |  |
| --- | --- |
| pic\_parameter\_set\_rbsp( ) { | Descriptor |
| **pic\_parameter\_set\_id** | ue(v) |
| **seq\_parameter\_set\_id** | ue(v) |
| **entropy\_coding\_synchro** | u(v) |
| **cabac\_istate\_reset\_flag** | u(1) |
| if( entropy\_coding\_synchro ) |  |
| **num\_substreams\_minus1** | ue(v) |
| **num\_temporal\_layer\_switching\_point\_flags** | ue(v) |
| for( i = 0; i < num\_temporal\_layer\_switching\_point\_flags; i++ ) |  |
| **temporal\_layer\_switching\_point\_flag**[ i ] | u(1) |
| **num\_ref\_idx\_l0\_default\_active\_minus1** | ue(v) |
| **num\_ref\_idx\_l1\_default\_active\_minus1** | ue(v) |
| [Ed. (BB): not present in HM software, depends on reference list construction decision ] |  |
| **pic\_init\_qp\_minus26** **/**\* relative to 26 \*/ | se(v) |
| [Ed. (BB): not present in HM software, signaled in slice header as absolute value slice\_qp, should be implemented to be used for slice\_qp\_delta] |  |
| **constrained\_intra\_pred\_flag** | u(1) |
| **slice\_granularity** | u(2) |
| **max\_cu\_qp\_delta\_depth** | ue(v) |
| [Ed. (BB): not present in HM software, should be implemented ] |  |
| **weighted\_pred\_flag** | u(1) |
| **weighted\_bipred\_idc** | u(2) |
| **tile\_info\_present\_flag** | u(1) |
| if( tile\_info\_present\_flag = = 1 **) {** |  |
| **num\_tile\_columns\_minus1** | ue(v) |
| **num\_tile\_rows\_minus1** | ue(v) |
| if( num\_tile\_columns\_minus1 != 0 | | num\_tile\_rows\_minus1 != 0 ) { |  |
| **tile\_boundary\_independence\_flag** | u(1) |
| **uniform\_spacing\_flag** | u(1) |
| if( !uniform\_spacing\_flag ) { |  |
| for( i = 0; i < num\_tile\_columns\_minus1; i++ ) |  |
| **column\_width[**i**]** | ue(v) |
| for( i = 0; i < num\_tile\_rows\_minus1; i++ ) |  |
| **row\_height[**i**]** | ue(v) |
| } |  |
| } |  |
| **}** |  |
| **bypass\_coding\_mode\_enabled\_flag** | u(1) |
| rbsp\_trailing\_bits( ) |  |
| } |  |

In 7.4.2.2 add

**bypass\_coding\_mode\_enabled\_flag** equal to 1 specifies that decoding process of inverse quantization, inverse transform, de-blocking filter, sample adaptive offset and adaptive loop filter is bypassed, and quantization scale parameter is not present in the picture.

Replace 8.3.3.1.6

##### Specification of Intra\_Angular prediction mode

With (changes are marked in yellow)

Inputs to this process are:

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

– a variable nS specifying the prediction size.

– if bypass\_coding\_mode\_enabled\_flagis equal to 1, residual samples residualSamples[x, y], with x, y = 0..nS-1;

Output of this process is:

– predicted samples predSamples[ x, y ], with x, y =0..nS-1.

This intra prediction mode is invoked when intraPredMode is in the range of 3..33.

Table 8‑6 specifies the mapping table between intraPredMode and the rearranged intra prediction order intraPredOrder.

Table 8‑7 – Specification of intraPredOrder

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

Figure 8‑2 illustrates the total 34 intra angles and Table 8‑7 specifies the mapping table between intraPredOrder and the angle parameter intraPredAngle.



Figure ‑ – Intra prediction angle definition (informative)

Table 8‑8 – Specification of intraPredAngle

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

Table 8‑8 further specifies the mapping table between intraPredOrder and the inverse angle parameter invAngle.

Table 8‑9 – Specification of invAngle

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **intraPredOrder** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **invAngle** | -256 | -315 | -390 | -482 | -630 | -910 | -1638 | -4096 |
| **intraPredOrder** | **18** | **19** | **20** | **21** | **22** | **23** | **24** | **25** |
| **invAngle** | -315 | -390 | -482 | -630 | -910 | -1638 | -4096 | - |

If bypass\_coding\_mode\_enabled\_flagis equal to 0, the reference pixel array refMain[ x ], with x=-nS..2\*nS is specified as follows.

– If intraPredOrder is less than 18,

refMain[ x ] = p[ -1+x, -1 ], with x=0..nS (8‑36)

* If intraPredAngle is less than 0,

refMain[ x ] = p[ -1, -1+( ( x\*invAngle+128 )>>8 ) ], with x=( nS\*intraPredAngle ) >>5..-1 (8‑37)

* Otherwise,

refMain[ x ] = p[ -1+x, -1 ], with x=nS+1..2\*nS (8‑38)

Otherwise,

refMain[ x ] = p[ -1, -1+x ], with x=0..nS (8‑39)

* If intraPredAngle is less than 0,

refMain[ x ] = p[ -1+( ( x\*invAngle+128 )>>8 ), -1 ], with x=( nS\*intraPredAngle ) >>5..-1 (8‑40)

* Otherwise,

refMain[ x ] = p[ -1, -1+x ], with x=nS+1..2\*nS (8‑41)

If bypass\_coding\_mode\_enabled\_flagis equal to 0, the values of the prediction samples predSamples[ x, y ], with x, y = 0..nS-1 are derived by the following procedures.

– The index variable iIdx and the multiplication factor iFact are derived by

iIdx = ( ( y + 1 )\*intraPredAngle ) >> 5 (8‑42)

iFact = ( ( y + 1 )\*intraPredAngle ) && 31 (8‑43)

– 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 )\*refMain[ x+iIdx+1 ] + iFact\*refMain[ x+iIdx+2] + 16 ) >> 5 (8‑44)

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

predSamples[ x, y ] = refMain[ x+iIdx+1 ] (8‑45)

If bypass\_coding\_mode\_enabled\_flagis equal to 1, the values of the prediction samples predSamples[ x, y ], with x, y = 0..nS-1 are derived by the following procedures.

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

– The multiplication factor iFact are derived by

iFact = ( intraPredAngle ) & 31 (8‑43)

– 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 )\*refMain[ 0 ] + iFact\*refMain[ 1] + 16 ) >> 5 (8‑44)

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

predSamples[ x, y ] = refMain[ 0 ] (8‑45)

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