|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  6th Meeting: Torino, IT, 14-22 July, 2011 | Document: JCTVC-F075 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **Unified scaling with adaptive offset for reference frame compression** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Dzung Hoang | Tel: Email: | [dzung.hoang@zenverge.com](mailto:dzung.hoang@zenverge.com) |
| *Source:* | Zenverge, Inc. | | |

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

# Abstract

Internal Bit Depth Increase (IBDI) is a technique that can often yield improved coding efficiency by increasing the arithmetic precision of the predictors, transforms, and loop filters in a video codec. The main drawback is an increase in memory storage and bandwidth requirements. Several reference frame compression techniques have been proposed to reduce the memory penalty of IBDI. This document describes two lightweight compression algorithms that operate on 4×4 blocks and can achieve compression ratios from 9:8 to 14:8. Unified Scaling with Adaptive Offset, the simpler of the two, introduces an average coding loss of about 0.5% for LB-HE and 0.1% for RA-HE. Unified Scaling with Adaptive Offset and DPCM, the better performing algorithm, reduces the coding loss to 0.3% for LB-HE and 0.1% for RA-HE.

# Introduction

Internal Bit Depth Increase (IBDI) [1] is a technique that can often yield improved coding efficiency by increasing the arithmetic precision of the predictors, transforms, and loop filters in a video codec. The main benefit of IBDI is additional coding gain due to improved intra-prediction and inter-prediction. The main drawback is increased memory storage and bandwidth requirements. There have been several proposals [4]–[7] to reduce the memory penalty of IBDI by compressing the reference frames before they are stored to the decoder picture buffers and decompressing them as needed for inter-prediction.

In this document, we present new experimental results using HM-3.0 of the Unified Scaling with Adaptive Offset (USAO) algorithm originally proposed in JCTVC-D035 [8]. We highlight the fact that USAO can support internal bit depth from 9 bits up to 15 bits, which is a desirable feature for adoption into the HEVC standard. We present experimental results for internal bit depth from 10 bits to 12 bits. Due to limitations of HM-3.0, we were unable to obtain results for higher internal bit depths.

Unified Scaling with Adaptive Offset is based upon Toshiba’s Adaptive Scaling (TAS) originally described in and recently extended in . We now briefly describe TAS and USAO. We then extend USAO with DPCM coding of the residual to improve its coding performance.

# Toshiba’s Adaptive Scaling

Toshiba’s Adaptive Scaling was originally proposed in to be used in conjunction with IBDI to mitigate the latter’s impact on memory storage and bandwidth due to the increase in internal pixel bit depth. Reference frame compression (RFC) is applied to each reference frame prior to storage in the decoded picture buffer, as depicted in .

|  |
| --- |
|  |

Figure : Codec diagram with IBDI and Reference Frame Compression [3]

TAS compression is performed on 4×4 blocks according to the pseudocode below, which is given for the case of 10-to-8 compression ratio according to .

Listing : TAS compression algorithm

|  |
| --- |
| M = (minimum pixel value in block) & 7) |
| R = (maximum pixel value in block) − M |
| for (S=0; R≥(128 << S) && S<2; S++); |
| if (S == 2) { |
| P0\_flag = (pixel\_value[0] < 2) ? 1 : (pixel\_value[0]+2)>>2 |
| P[i] = min(255, (pixel\_value[i] + 2) >> 2), for all i > 0 |
| } else { |
| P0\_flag = 0 |
| P[i] = (pixel\_value[i] − M) >> S, for all i |
| } |

An example compression format for TAS is shown in Listing 2.

Listing : TAS compression storage format

|  |
| --- |
| u(8) /\* P0\_flag \*/ |
| if (P0\_flag != 0) { |
| for (i=1; i<16; i++) u(8) /\* pixel value P[i] \*/ |
| } else { |
| u(1) /\* S: [0..1] \*/ |
| u(7) /\* M >> 3 \*/ |
| for (i=0; i<16; i++) u(7) /\* pixel value P[i] \*/ |
| } |

The TAS decompression algorithm is presented in pseudocode in Listing 3.

Listing : TAS decompression algorithm

|  |
| --- |
| if (P0\_flag == 0) { |
| D[i] = (P[i] << S) + M + (S == 0 ? 0 : 1), for all i |
| } else { |
| D[0] = (P0\_flag << 2) |
| D[i] = P[i] << 2, for all i > 0 |
| } |

TAS uses a form of escape coding to signal between two compression techniques: fixed rounding and adaptive scaling. serves two purposes. When , adaptive scaling is used. When , fixed rounding is used and the value of is used to reconstruct pixel\_value[0].



With fixed rounding, each increased-bit-depth pixel value is quantized to 8 bits with rounding. Reconstruction is performed with a simple left shift.

With adaptive scaling, the minimum pixel value is quantized using a fixed quantizer and the quantized minimum is used to predict the pixel values in the block. The residual values are then quantized by right-shifting by . Reconstruction is performed with an additive offset of 1 when . This can be generalized to an additive offset of when .



# Unified Scaling with Adaptive Offset

With USAO, only the adaptive scaling mode of TAS is used. In addition, an adaptive offset is used in reconstruction versus the fixed offset used in TAS.

The USAO compression algorithm is described in pseudocode in Listing 4. It can be shown that the floating-point reconstruction offset that minimizes the mean square error can be computed as the average of the quantization error. With a fixed-point offset, it was empirically found that a rounding constant of 7 gives the same results as a brute force search. (In [8], a rounding constant of 8 was used.)

Listing : USAO compression algorithm

|  |
| --- |
| M = (minimum pixel value in block) |
| R = (maximum pixel value in block) |
| for (S=0; (R>>S)−(M>>S)≥128; S++); |
| mask = (1 << S) − 1 |
| offset = (sum(pixel\_value[i] & mask) + 7) / 16 |
| M = M & ~mask |
| P[i] = (pixel\_value[i] − M) >> S, for all i |

The USAO decompression algorithm is described in Listing 5.

Listing : USAO decompression algorithm

|  |
| --- |
| D[i] = (P[i] << S) + M + offset, for all i |

Let denote the internal bit depth. Although Listing 4 does not refer to explicitly, the effect of is to limit the maximum value that can take. The objective of the scaling factor is to reduce the dynamic range of the residuals in a block down to 7 bits. Therefore the maximum value for given -bit input is . The range for is thus . If we code in binary, the number of bits required would be . This is shown in Table 1 for several values of .



Table : Number of bits to represent S in binary

|  |  |  |
| --- | --- | --- |
|  | range of | bits to code |
| 9 |  | 2 |
| 10 |  | 2 |
| 11 |  | 3 |
| 12 |  | 3 |
| 13 |  | 3 |
| 14 |  | 3 |
| 15 |  | 4 |

An example compression format for USAO is presented in Listing 6. This format uses bits. The number of bits is tabulated against in Table 2. Since the quantized minimum pixel value is already coded as , we can save 3 bits by coding the index of and skip the coding of , which would otherwise be 0.



Listing : Example compression format for USAO

|  |
| --- |
| u() /\* S \*/ |
| u() /\* M >> S \*/ |
| u() /\* offset \*/ |
| u(4) /\* M\_index \*/ |
| for (i=0; i<15; i++) u(7) /\* P array excluding M \*/ |

Table : Number of bits used by USAO compression format

|  |  |
| --- | --- |
|  | bits used per 4×4 block |
| 9 | 120 |
| 10 | 121 |
| 11 | 123 |
| 12 | 124 |
| 13 | 125 |
| 14 | 126 |
| 15 | 128 |

# Unified Scaling with Adaptive Offset and DPCM

We now extend USAO with a more efficient coding of the residual array . With USAO, the residual values are computed as the difference between the original pixel values and the quantized minimum : . The scale factor is computed such that the residual for the maximum pixel value can be represented using 7 bits. For blocks where the residual changes gradually, the differences between adjacent residuals can often be coded with fewer bits for a given scale factor using DPCM. Oftentimes, a smaller scale factor can be used with DPCM, thereby lowering the distortion.



We call this algorithm Unified Scaling with Adaptive Offset and DPCM (USAO-D).

## DPCM Coding Outline

DPCM coding of the residual array consists of the following steps: scanning and delta coding.

### Scanning Pattern

With differential coding, the order in which the differences are computed is an important consideration. A scan pattern determines the order in which the differences are computed. A scan pattern should cover every sample location exactly once. Given that the index of the minimum sample is coded in the optimized compression format given in , it makes sense to start the scan at the location of the minimum sample. Given that the minimum sample can occur at any location, the scanning pattern must be cyclic. In a traditional scan pattern such as the zig-zag scan, the pattern begins at the upper-left sample and ends at the lower-right sample . Making the zig-zag scan into a cyclic scan would introduce a jump between and . A better scanning pattern would have the property that successive positions in the scan map to adjacent spatial positions. Below in are three examples of cyclic scans that have this property. Rotations and flips of these basic patterns can also be considered. In addition, many other cyclic scan patterns are possible. After some initial experiments, we chose to use the scanning pattern in a.



|  |  |  |
| --- | --- | --- |
|  |  |  |
| a | b | c |

Figure : Examples of 4×4 cyclic scans with adjacency property

### Delta Coding

With Zenverge Unified Scaling, we can code the residual using unsigned 7-bit integers because the residual is computed against the minimum sample. With DPCM coding, the delta values must be represented as signed 7-bit integers. If we use twos-complement representation, a 7-bit integer can represent the integers between −64 and 63, inclusive. However, if the previous pixel in scan order is the minimum pixel, for example, then using twos-complement is inefficient because half of the range is not being used. Instead, we can use excess- representation, where is adapted per coded sample to maximize the range of representable differences. The same technique can be applied when the previous pixel is near the maximum value. The minimum pixel value is already signaled as , so we need to signal the maximum if we wish to increase the range of representable differences. Due to a limited bit budget, instead of coding the maximum, we can code the difference between the maximum and minimum, which we call the range. The range can be further quantized to reduce the number of bits needed to code it.



Here is an outline of how delta coding works with adaptive excess- coding of the differences.



Listing : Outline of delta coding

|  |
| --- |
| Assume S, coded\_min, and range have been computed. |
| coded\_max = coded\_min + range |
| Loop over 15 samples in scan order starting with sample that follows the minimum sample in scan order: |
| cur\_pel = (sample at current scan position) >> S |
| prev\_pel = (sample at previous scan position) >> S |
| min\_delta = −64 |
| if (coded\_min − prev\_ pel > −64) { |
| min\_delta = coded\_min − prev\_pel |
| } else if (coded\_max − prev\_ pel < 63) { |
| min\_delta = coded\_max − prev\_pel − 127 |
| } |
| delta = (cur\_pel − prev\_pel − min\_delta) |

## Signaling DPCM Mode and Coding Range

With delta coding, the scaling factor needs to be chosen to limit delta to within the range . It is possible to need to use a higher scaling factor when using DPCM mode. Therefore we need to signal when to use DPCM. Since the range is only needed in DPCM mode, we can combine the signaling of DPCM mode with the coding of the range into one codeword called dpcm\_flag\_range. Let denote the number of bits available to code dpcm\_flag\_range. We use these bits to code the -MSB of range. We reserve the value 0 to signal not to use DPCM.



We quantize the range by simply keeping the -MSB. This is equivalent to a right shift by . Because the range is used to code a maximum value, when we decode (inverse quantize) the range, we do not simply perform a left shift by . In addition, we need to fill in the least significant bits with 1’s. This step makes sure that the decoded maximum value is indeed higher than any pixel value. An efficient way to accomplish this is as follows. Let denote the quantized range. The inverse quantized range can be computed as .



## Compression Format

We now define a nominal compression format for USAO-D. This compression format is only to show that the compressed data can fit in 128 bits. It should not be construed as a normative requirement.

Listing : Compression format for USAO-D

|  |
| --- |
| u(k) /\* dpcm\_flag\_range \*/ |
| u( ceil(log2(N−6)) ) /\* S \*/ |
| u(N−S) /\* M \*/ |
| u(S) /\* offset \*/ |
| u(4) /\* M\_index \*/ |
| for (i=0; i<15; i++) u(7) /\* P array excluding M \*/ |

From this table, we can compute the number of bits used to code each block as follows:



The value of for each value of, given a budget of 128 bits, are tabulated below. Note that with , there are not enough available bits to code dpcm\_flag\_range. Therefore USAO-D is limited to .



Table : Number of bits to code and dpcm\_flag\_range



|  |  |  |
| --- | --- | --- |
|  | **bits for** | **=bits for dpcm\_flag\_range** |
| 9 | 2 | 8 |
| 10 | 2 | 7 |
| 11 | 3 | 5 |
| 12 | 3 | 4 |
| 13 | 3 | 3 |
| 14 | 3 | 2 |
| 15 | 4 | 0 |

## Compression Algorithm

In the compression algorithm, we need to determine both the scaling factor and whether to use DPCM. We first compute for the non-DPCM case as in . Then we see if we can use a smaller value of with DPCM. Pseudocode for the compression algorithm is given in .



Listing : Pseudocode for USAO-D compression

|  |
| --- |
| M = (minimum pixel value in block) |
| M\_index = least position of minimum pixel value |
| R = (maximum pixel value in block) |
| for (S=0; (R>>S)−(M>>S)>=128; S++); |
| valid = 0 |
| for (Sd=0; Sd<S && !valid; Sd += !valid) { |
| coded\_min = M >> Sd |
| dpcm\_flag\_range = max( 1, (R − M) >> (N − k) ) |
| coded\_max = coded\_min + (((dpcm\_flag\_range+1) << (N − k)) >> Sd) − 1 |
| start scan at position after M\_index |
| for (i=0; i<15; i++) { |
| cur\_pel = (sample at current scan position) >> Sd |
| prev\_pel = (sample at previous scan position) >> Sd |
| min\_delta = −64 |
| if (coded\_min − prev\_ pel > −64) { |
| min\_delta = coded\_min − prev\_pel |
| } else if (coded\_max − prev\_ pel < 63) { |
| min\_delta = coded\_max − prev\_pel − 127 |
| } |
| P[i] = (cur\_pel − prev\_pel − min\_delta) |
| go to next scan position |
| } |
| if (0 ≤ P[i] < 128, for all i) |
| valid = 1 |
| } |
| if (Sd < S) then |
| S = Sd |
| else |
| dpcm\_flag\_range = 0 |
| mask = (1 << S) − 1 |
| offset = ( sum( pixel\_value[i] & mask ) + 7 ) >> 4 |
| M = M & ~mask |
| if (dpcm\_flag\_range == 0) |
| P[i] = (pixel\_value[i] − M) >> S, for all i |

## Decompression Algorithm

The decompression algorithm is described in pseudocode below.

Table : Pseudocode for USAO-D decompression

|  |
| --- |
| if (dpcm\_flag\_range == 0) then { |
| P[M\_index] = 0 |
| D[i] = (P[i] << S) + M + offset, for all i |
| } else { |
| coded\_min = M >> S |
| coded\_max = coded\_min + (((dpcm\_flag\_range+1) << (N − k)) >> S) − 1 |
| D[M\_index] = M + offset |
| start scan at position after M\_index |
| for (i=0; i<15; i++) { |
| prev\_pel = (D[ previous scan position]) >> S |
| min\_delta = −64 |
| if (coded\_min − prev\_ pel > −64) { |
| min\_delta = coded\_min − prev\_pel |
| } else if (coded\_max − prev\_ pel < 63) { |
| min\_delta = coded\_max − prev\_pel − 127 |
| } |
| D[current scan position] = ((P[i] + min\_delta + prev\_pel) << S) + offset |
| go to next scan position |
| } |
| } |

# Experimental Results

USAO and USAO-D were implemented on top of the HM-3.0-svn-memory version of HM-3.0 that integrates the decoder-side memory access measurement module. HM-3.0-svn-memory was used to generate the anchor bitstreams.

## BD-Rate Results for USAO

Simulations were run with internal bit depths of 10, 11, and 12. As proscribed by the AHG on Memory Compression, only the LB-HE and RA-HE common conditions were used. BD-rate results are listed below in to . The overall coding loss of USAO averages between 0.43% to 0.48% for LB-HE and about 0.1% for RA-HE across all tested internal bit depths. Encoder run times for USAO is about 100% to 101% of HM, which is within measurement tolerance. Decoder run times are between 101% and 104%.

Table : USAO versus HM for internal bit depth of 10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 10 | | | Internal Bit Depth = 10 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | 0.00 | -0.13 | -0.26 | | Class B | 0.28 | 0.23 | 0.15 | 0.10 | -0.09 | -0.04 | | Class C | 0.21 | 0.22 | 0.10 | 0.12 | 0.07 | 0.02 | | Class D | 0.17 | 0.38 | 0.64 | 0.14 | 0.00 | -0.03 | | Class E | 1.61 | 0.24 | 0.66 |  |  |  | | Overall | 0.48 | 0.27 | 0.36 | 0.09 | -0.04 | -0.07 | | Enc Time[%] | 101% | | | 100% | | | | Dec Time[%] | 104% | | | 103% | | | |

Table : USAO versus HM for internal bit depth of 11

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 11 | | | Internal Bit Depth = 11 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | 0.02 | -0.24 | 0.23 | | Class B | 0.28 | 0.18 | 0.28 | 0.10 | 0.03 | 0.05 | | Class C | 0.22 | 0.17 | 0.31 | 0.11 | 0.19 | 0.10 | | Class D | 0.16 | 0.29 | 0.01 | 0.15 | 0.01 | 0.12 | | Class E | 1.41 | 0.16 | 0.49 |  |  |  | | Overall | 0.45 | 0.20 | 0.26 | 0.09 | 0.00 | 0.12 | | Enc Time[%] | 100% | | | 100% | | | | Dec Time[%] | 104% | | | 103% | | | |

Table : USAO versus HM for internal bit depth of 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 12 | | | Internal Bit Depth = 12 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | 0.10 | 0.08 | -0.23 | | Class B | 0.27 | 0.07 | 0.39 | 0.11 | 0.08 | 0.09 | | Class C | 0.23 | 0.10 | 0.23 | 0.15 | 0.06 | 0.18 | | Class D | 0.17 | 0.37 | 0.06 | 0.14 | 0.06 | 0.04 | | Class E | 1.32 | 0.61 | 0.50 |  |  |  | | Overall | 0.43 | 0.25 | 0.29 | 0.12 | 0.07 | 0.03 | | Enc Time[%] | 101% | | | 100% | | | | Dec Time[%] | 104% | | | 101% | | | |

It is instructive to also compare USAO with IBDI against HM without IBDI to see the gains that can be achieved with the same memory footprint. The results of this comparison are given in to . The gains with IBDI and USAO-D average between 2.2% to 2.4% for LB-HE and 1.5% for RA-HE across all tested internal bit depths. Encoder run times are between 100% and 101%. Decoder run times are between 105% to 107%.

Table : USAO versus HM w/o IBDI for internal bit depth of 10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 10 | | | Internal Bit Depth = 10 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -1.41 | -4.58 | -5.08 | | Class B | -2.64 | -9.59 | -10.38 | -2.44 | -7.61 | -6.93 | | Class C | -1.30 | -4.80 | -5.59 | -1.24 | -3.74 | -3.88 | | Class D | -0.71 | -7.29 | -7.24 | -0.79 | -2.99 | -2.98 | | Class E | -4.80 | -14.75 | -11.07 |  |  |  | | Overall | -2.23 | -8.78 | -8.53 | -1.54 | -4.94 | -4.82 | | Enc Time[%] | 101% | | | 100% | | | | Dec Time[%] | 105% | | | 106% | | | |

Table : USAO versus HM w/o IBDI for internal bit depth of 11

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 11 | | | Internal Bit Depth = 11 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -1.43 | -4.79 | -5.24 | | Class B | -2.71 | -10.23 | -11.08 | -2.52 | -7.95 | -6.95 | | Class C | -1.29 | -5.22 | -5.86 | -1.27 | -3.83 | -4.06 | | Class D | -0.69 | -7.67 | -7.83 | -0.73 | -3.15 | -3.32 | | Class E | -5.60 | -15.38 | -12.72 |  |  |  | | Overall | -2.39 | -9.30 | -9.27 | -1.57 | -5.15 | -4.98 | | Enc Time[%] | 101% | | | 101% | | | | Dec Time[%] | 105% | | | 107% | | | |

Table : USAO versus HM w/o IBDI for internal bit depth of 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 12 | | | Internal Bit Depth = 12 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -1.39 | -4.76 | -5.52 | | Class B | -2.65 | -10.42 | -11.34 | -2.49 | -8.09 | -7.19 | | Class C | -1.23 | -5.24 | -5.88 | -1.21 | -3.79 | -4.07 | | Class D | -0.54 | -7.74 | -7.84 | -0.74 | -3.03 | -3.17 | | Class E | -5.93 | -14.50 | -13.12 |  |  |  | | Overall | -2.38 | -9.22 | -9.43 | -1.53 | -5.15 | -5.06 | | Enc Time[%] | 101% | | | 101% | | | | Dec Time[%] | 105% | | | 107% | | | |

## Memory Bandwidth Reduction for USAO

The memory bandwidth reduction achieved with USAO compared to HM-3.0 is tabulated below for internal bit depth of 10.

Table : Memory bandwidth comparison between USAO and HM for LB-HE

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LB-HE | **Memory bandwith increase %** | | | | | | | |
|  | **8bit/8bit** | **32bit/ 64bit** | **32bit/128bit** | **64bit/128bit** | **64bit/256bit** | **64bit/512bit** | **64bit/256bit FIFO** | **64bit/512bit FIFO** |
| Class A |  |  |  |  |  |  |  |  |
| Class B | 0.55% | -10.91% | -18.82% | -21.95% | -26.41% | -41.43% | -29.75% | -33.16% |
| Class C | 4.58% | -9.78% | -20.46% | -24.70% | -31.11% | -48.09% | -38.37% | -42.85% |
| Class D | 12.63% | -5.81% | -18.85% | -24.01% | -31.63% | -50.26% | -39.34% | -44.56% |
| Class E | -11.88% | -16.70% | -21.58% | -23.12% | -33.11% | -48.43% | -30.85% | -33.44% |
| All | 2.25% | -10.44% | -19.75% | -23.37% | -30.15% | -46.62% | -34.51% | -38.49% |

Table : Memory bandwidth comparison between USAO and HM for RA-HE

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RA-HE | **Memory bandwith increase %** | | | | | | | |
|  | **8bit/8bit** | **32bit/ 64bit** | **32bit/128bit** | **64bit/128bit** | **64bit/256bit** | **64bit/512bit** | **64bit/256bit FIFO** | **64bit/512bit FIFO** |
| Class A | -2.54% | -13.02% | -20.11% | -22.67% | -26.78% | -39.45% | -28.31% | -30.66% |
| Class B | -2.66% | -12.61% | -19.13% | -21.61% | -25.35% | -40.15% | -26.73% | -30.20% |
| Class C | 1.82% | -10.70% | -19.82% | -23.58% | -29.05% | -44.83% | -34.45% | -38.49% |
| Class D | 8.08% | -7.16% | -18.41% | -23.19% | -29.63% | -46.88% | -35.73% | -40.00% |
| Class E |  |  |  |  |  |  |  |  |
| All | 0.89% | -11.03% | -19.44% | -22.76% | -27.71% | -42.81% | -31.11% | -34.65% |

## BD-Rate Results for USAO-D

Simulations were run with internal bit depths of 10, 11, and 12. As proscribed by the AHG on Memory Compression, only the LB-HE and RA-HE common conditions were used. BD-rate results are listed below in to . The overall coding loss of USAO-D averages 0.3% for LB-HE and 0.1% for RA-HE across all tested internal bit depths. Encoder run times for USAO-D is about 99% of HM, which is within measurement tolerance. Decoder run times are between 102% and 105%.

Table : USAO-D versus HM for internal bit depth of 10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 10 | | | Internal Bit Depth = 10 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | 0.00 | -0.22 | 0.05 | | Class B | 0.20 | 0.24 | 0.30 | 0.08 | -0.04 | 0.01 | | Class C | 0.18 | 0.18 | 0.07 | 0.08 | 0.07 | 0.01 | | Class D | 0.12 | 0.08 | 0.69 | 0.17 | -0.01 | -0.03 | | Class E | 1.11 | 0.31 | 0.08 |  |  |  | | Overall | 0.35 | 0.20 | 0.30 | 0.08 | -0.05 | 0.01 | | Enc Time[%] | 99% | | | 98% | | | | Dec Time[%] | 104% | | | 104% | | | |

Table : USAO-D versus HM for internal bit depth of 11

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 11 | | | Internal Bit Depth = 11 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -0.01 | 0.07 | 0.01 | | Class B | 0.21 | 0.10 | 0.12 | 0.07 | 0.01 | -0.01 | | Class C | 0.16 | 0.07 | 0.29 | 0.12 | 0.15 | 0.09 | | Class D | 0.14 | 0.00 | 0.04 | 0.16 | 0.11 | 0.05 | | Class E | 1.09 | 0.25 | 0.61 |  |  |  | | Overall | 0.34 | 0.09 | 0.24 | 0.08 | 0.08 | 0.03 | | Enc Time[%] | 99% | | | 99% | | | | Dec Time[%] | 105% | | | 103% | | | |

Table : USAO-D versus HM for internal bit depth of 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 12 | | | Internal Bit Depth = 12 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | 0.09 | 0.21 | 0.11 | | Class B | 0.18 | 0.21 | 0.15 | 0.08 | 0.01 | 0.03 | | Class C | 0.20 | 0.05 | 0.01 | 0.11 | 0.06 | 0.19 | | Class D | 0.14 | 0.30 | 0.42 | 0.15 | 0.10 | -0.04 | | Class E | 0.98 | 0.03 | 0.32 |  |  |  | | Overall | 0.33 | 0.16 | 0.22 | 0.11 | 0.09 | 0.07 | | Enc Time[%] | 99% | | | 99% | | | | Dec Time[%] | 105% | | | 102% | | | |

It is instructive to also compare USAO-D with IBDI against HM without IBDI to see the gains that can be achieved with the same memory footprint. The results of this comparison are given in to . The gain with IBDI and USAO-D averages between -2.4% to -2.5% for LB-HE and -1.6% for RA-HE across all tested internal bit depths. Encoder run times are between 99% and 100%. Decoder run times are between 106% to 108%.

Table : USAO-D versus HM w/o IBDI for internal bit depth of 10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 10 | | | Internal Bit Depth = 10 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -1.40 | -4.60 | -4.96 | | Class B | -2.72 | -9.57 | -10.21 | -2.47 | -7.56 | -6.89 | | Class C | -1.33 | -4.83 | -5.61 | -1.28 | -3.74 | -3.89 | | Class D | -0.75 | -7.59 | -7.24 | -0.76 | -2.99 | -2.98 | | Class E | -5.27 | -14.69 | -11.61 |  |  |  | | Overall | -2.36 | -8.85 | -8.58 | -1.55 | -4.93 | -4.79 | | Enc Time[%] | 100% | | | 99% | | | | Dec Time[%] | 106% | | | 107% | | | |

Table : USAO-D versus HM w/o IBDI for internal bit depth of 11

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 11 | | | Internal Bit Depth = 11 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -1.45 | -4.75 | -5.22 | | Class B | -2.79 | -10.30 | -11.28 | -2.56 | -7.97 | -7.02 | | Class C | -1.35 | -5.33 | -5.89 | -1.26 | -3.87 | -4.07 | | Class D | -0.70 | -7.92 | -7.83 | -0.72 | -3.05 | -3.39 | | Class E | -5.91 | -15.24 | -12.60 |  |  |  | | Overall | -2.49 | -9.39 | -9.32 | -1.57 | -5.14 | -5.02 | | Enc Time[%] | 100% | | | 99% | | | | Dec Time[%] | 107% | | | 107% | | | |

Table : USAO-D versus HM w/o IBDI for internal bit depth of 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Internal Bit Depth = 12 | | | Internal Bit Depth = 12 | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -1.41 | -4.78 | -5.43 | | Class B | -2.74 | -10.28 | -11.56 | -2.52 | -8.16 | -7.24 | | Class C | -1.26 | -5.28 | -6.09 | -1.24 | -3.80 | -4.06 | | Class D | -0.57 | -7.81 | -7.51 | -0.73 | -2.99 | -3.25 | | Class E | -6.26 | -15.11 | -13.34 |  |  |  | | Overall | -2.48 | -9.32 | -9.51 | -1.55 | -5.16 | -5.09 | | Enc Time[%] | 99% | | | 99% | | | | Dec Time[%] | 107% | | | 108% | | | |

## Memory Bandwidth Reduction for USAO-D

The memory bandwidth reduction achieved with USAO-D compared to HM-3.0 is tabulated below for internal bit depth of 10.

Table : Memory bandwidth comparison between USAO-D and HM for LB-HE

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LB-HE | **Memory bandwith increase %** | | | | | | | |
|  | **8bit/8bit** | **32bit/ 64bit** | **32bit/128bit** | **64bit/128bit** | **64bit/256bit** | **64bit/512bit** | **64bit/256bit FIFO** | **64bit/512bit FIFO** |
| Class A |  |  |  |  |  |  |  |  |
| Class B | 0.50% | -10.95% | -18.85% | -21.98% | -26.44% | -41.47% | -29.82% | -33.22% |
| Class C | 4.55% | -9.80% | -20.48% | -24.72% | -31.13% | -48.11% | -38.36% | -42.84% |
| Class D | 12.51% | -5.91% | -18.93% | -24.09% | -31.70% | -50.32% | -39.23% | -44.45% |
| Class E | -12.07% | -16.88% | -21.76% | -23.29% | -33.27% | -48.55% | -31.07% | -33.65% |
| All | 2.16% | -10.51% | -19.82% | -23.44% | -30.21% | -46.67% | -34.54% | -38.51% |

Table : Memory bandwidth comparison between USAO-D and HM for RA-HE

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RA-HE | **Memory bandwith increase %** | | | | | | | |
|  | **8bit/8bit** | **32bit/ 64bit** | **32bit/128bit** | **64bit/128bit** | **64bit/256bit** | **64bit/512bit** | **64bit/256bit FIFO** | **64bit/512bit FIFO** |
| Class A | -2.59% | -13.07% | -20.15% | -22.72% | -26.82% | -39.48% | -28.32% | -30.66% |
| Class B | -2.69% | -12.64% | -19.16% | -21.64% | -25.38% | -40.17% | -26.76% | -30.23% |
| Class C | 1.79% | -10.73% | -19.85% | -23.61% | -29.07% | -44.85% | -34.45% | -38.49% |
| Class D | 8.05% | -7.19% | -18.44% | -23.22% | -29.66% | -46.92% | -35.73% | -40.01% |
| Class E |  |  |  |  |  |  |  |  |
| All | 0.86% | -11.06% | -19.47% | -22.79% | -27.73% | -42.84% | -31.12% | -34.66% |

# Comparison with Other Proposals

## Toshiba JCTVC-F319

We compare Toshiba’s Adaptive Scaling with Offset (TASO) proposed in with our USAO and USAO-D. The results indicate that USAO has a 0.27% coding gain for LB-HE configuration and that USAO-D has a 0.40% coding gain for LB-HE. The biggest differences can be attributed to the Class E sequences. For Class E LB-HE, USAO has a coding gain of 1.35% and USAO-D has a coding gain of 1.83%. We suspect that the difference due to TASO using fixed-rounding mode in some cases whereas USAO and USAO-D always use adaptive scaling mode.

Table : BD-Rate results comparing TASO (reference) against USAO (target)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -0.07 | -0.06 | -0.46 | | Class B | -0.11 | 0.12 | -0.08 | 0.00 | -0.05 | -0.05 | | Class C | -0.01 | 0.01 | -0.04 | 0.02 | -0.08 | -0.09 | | Class D | 0.08 | 0.21 | 0.11 | 0.06 | 0.01 | -0.06 | | Class E | -1.35 | -0.33 | -0.09 |  |  |  | | Overall | -0.27 | 0.03 | -0.03 | 0.00 | -0.05 | -0.16 | | Enc Time[%] | 99% | | | 100% | | | | Dec Time[%] | 100% | | | 98% | | | |

Table : BD-Rate results comparing TASO (reference) against USAO-D (target)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Y | U | V | Y | U | V | | Class A |  |  |  | -0.07 | -0.15 | -0.15 | | Class B | -0.19 | 0.13 | 0.06 | -0.03 | -0.01 | -0.01 | | Class C | -0.04 | -0.03 | -0.06 | -0.01 | -0.08 | -0.10 | | Class D | 0.04 | -0.10 | 0.16 | 0.09 | 0.01 | -0.07 | | Class E | -1.83 | -0.27 | -0.67 |  |  |  | | Overall | -0.40 | -0.04 | -0.08 | -0.01 | -0.06 | -0.08 | | Enc Time[%] | 98% | | | 99% | | | | Dec Time[%] | 101% | | | 99% | | | |

## MediaTek JCTVC-F073

We compare USAO and USAO-D against Version 2 of the proposal by MediaTek in JCTVC-F073 . Results are reported below. The reference and target software were run on very different clusters and therefore the run times are not reliable and should be ignored. The results indicate that USAO has a 0.19% luma coding loss but 0.01% U and 0.12% V gain for LB-HE configuration. For the RA-HE configuration, the luma coding loss of 0.10% is balanced against a coding gain of 0.02% for U and 0.20% for V.

For the LB-HE configuration, USAO-D has a 0.05% luma coding loss but 0.09% and 0.18% gains for U and V, respectively. For the RA-HE configuration, USAO-D has a coding loss of 0.09% and 0.01% for Y and U, resp., and a coding gain of 0.12% for V. It is very curious that the results for the Class E sequences are so close between USAO-D and JLCA Version 2. From experience and other results presented above, the Class E sequences are the most sensitive to RFC.

Table : BD-Rate results comparing JLCA Version 2 (reference) against USAO (target)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Y | U | V | Y | U | V | | Class A |  |  |  | 0.01 | -0.08 | -0.49 | | Class B | 0.09 | -0.02 | -0.28 | 0.10 | 0.01 | -0.08 | | Class C | 0.10 | -0.05 | -0.34 | 0.10 | -0.01 | -0.14 | | Class D | 0.17 | 0.13 | -0.23 | 0.18 | 0.15 | -0.12 | | Class E | 0.51 | -0.15 | 0.61 |  |  |  | | **Overall** | 0.19 | -0.01 | -0.12 | 0.10 | 0.02 | -0.20 | | Enc Time[%] | 106% | | | 108% | | | | Dec Time[%] | 155% | | | 157% | | | |

Table : BD-Rate results comparing JLCA Version 2 (reference) against USAO-D (target)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Low delay B HE | | | Random Access HE | | |
| Y | U | V | Y | U | V |
| Class A |  |  |  | 0.01 | -0.17 | -0.18 |
| Class B | 0.01 | -0.01 | -0.14 | 0.07 | 0.05 | -0.04 |
| Class C | 0.06 | -0.09 | -0.37 | 0.06 | -0.01 | -0.15 |
| Class D | 0.12 | -0.18 | -0.19 | 0.20 | 0.14 | -0.13 |
| Class E | 0.01 | -0.08 | 0.02 |  |  |  |
| Overall | 0.05 | -0.09 | -0.18 | 0.09 | 0.01 | -0.12 |
| Enc Time[%] | 105% | | | 106% | | |
| Dec Time[%] | 155% | | | 158% | | |

# Histogram of MSE for USAO-D

We gathered histogram data for block-level MSE to quantify the level of distortion introduced by USAO-D. We chose three bitstreams that exhibited relatively high coding loss: Cactus with LB-HE, Vidyo4 with LB-HE, and BQSquare with RA-HE. Tables and plots are shown below. The numbers for the plots can be examined by opening the embedded Excel charts or in the Excel file included in the ZIP file with this proposal.

The histograms show that the distortion is very well-controlled and should not introduce any noticeable artifacts. Because scalar quantization is used in this proposal, any visible artifacts would be such similar to contouring.

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

Figure : Histogram Plots of Block MSE for Selected Sequences

# Further Notes on USAO-D

We gathered some additional statistics on USAO-D and found that more than 99.99% of the time when DPCM mode is chosen, the difference in S is 1 compared to if DPCM mode were not chosen. Let S\* denote the scale factor to be used in non-DPCM mode. We can limit the search for S in DPCM mode only to S\*-1 to decrease the complexity.

# Conclusions

The coding gains that can be achieved by using IBDI make it a very attractive technique to be implemented in a video codec system. However, it comes with a cost of increased memory storage and bandwidth requirements for both encoders and decoders. Reference frame compression that is employed in the encoding loop can reduce the memory overhead of IBDI. In addition, RFC can also be applied to code video with bit depth higher than 8 while using the memory storage requirement for 8-bit video. We propose two lightweight compression algorithms that can maintain much of the gains of IBDI and can operate over a wide range of internal bit depths from 9 to 14. The two algorithms exhibit a tradeoff between complexity and coding performance. USAO, the simpler of the two, introduces an average coding loss of about 0.5% for LB-HE and 0.1% for RA-HE. USAO-D, the better performing algorithm, reduces the coding loss to 0.3% for LB-HE and 0.1% for RA-HE. On the Class E sequences, USAO-D reduces the coding loss to 1.1% from 1.6% achieved with USAO.

An RFC algorithm should be included in the HEVC standard to allow implementers the option to achieve coding gains (with IBDI) and support higher bit depth video without increasing the memory storage and bandwidth requirements beyond that needed for 8-bit video. USAO and USAO-D should be considered as candidates for standardization based upon their coding performance and low complexity.

# References

1. T. Chujoh and R. Noda, “Internal bit depth increase except frame memory,” ITU-T SG16 Q.6 Document, VCEG-AF07, San Jose, April 2007.
2. T. Chujoh, A. Tanizawa and T. Yamakage, “Description of video coding technology proposal by TOSHIBA,” JCT-VC Document, JCTVC-A117, Dresden, April 2010.
3. T. Chujoh, T. Shiodera and T. Yamakage, “TE2: Adaptive scaling for bit depth compression on IBDI,” JCT-VC Document, JCTVC-B044, Geneva, July 2010.
4. T. Chujoh and T. Yamakage, “Adaptive scaling for reference pictures memory compression,” JCT-VC Document JCTVC-E133, Geneva, March 2011.
5. H. Aoki, et. al., “TE2: 1-D DPCM-based memory compression,” JCT-VC Document JCTVC-C093, Guangzhou, October 2010.
6. M. Zhou and M. Budagavi, “TE2: TI reference frame compression proposal,” JCT-VC Document JCTVC-C060, Guangzhou, October 2010.
7. V. Wahadaniah, et. al., “TE2: Reference frame compression using image coder,” JCT-VC Document JCTVC-C073, Guangzhou, October 2010.
8. D. Hoang, “Unified scaling with adaptive offset for reference frame compression with IBDI,” JCT-VC Document JCTVC-D035, Daegu, January 2011.
9. T. Chujoh and T. Yamakage, “Adaptive scaling with offset for reference pictures memory compression,” JCT-VC Document JCTVC-F319, Torino, July 2011.
10. S. Liu, X. Zhang, and S. Lei, “Joing luma-chroma adaptive reference picture memory compression,” JCT-VC Document JCTVC-F073, Torino, July 2011.

# Patent rights declaration(s)

**Zenverge, 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).**