|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  6th Meeting: Torino, 14-22 July, 2011 | Document: JCTVC-F079 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | Crosscheck of JCTVC-F073 proposal for joint luma-chroma adaptive reference picture memory compression | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Information | | |
| *Author(s) or Contact(s):* | Dzung Hoang | Tel: Email: | +1-408-532-5694 dzung.hoang@zenverge.com |
| *Source:* | Zenverge, Inc. | | |

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

# Abstract

This document reviews MediaTek’s JCTVC-F073 proposal for joint luma-chroma adaptive reference picture memory compression. MediaTek’s BD-rate results comparing Version 1 and Version 2 of their algorithm against HM 3.0 are verified.

# Introduction

In JCTVC-F073 [1], MediaTek proposes several reference frame compression (RFC) schemes for 10-to-8 compression. The proposed schemes are based upon Toshiba’s Adaptive Scaling [2] and jointly allocate bits among co-located luma and chroma blocks to give more bits to the luma blocks and fewer bits to the chroma blocks. In addition, preliminary results for combining Zenverge’s Adaptive Offset [3] are presented. In this document we refer to the general scheme of joint luma-chroma adaptive reference picture compression as JLCA.

MediaTek shared with us an early version of their software that supports Version 1 and Version 2 of their proposed algorithms. This software used HM 3.0 with bug fix 146 as a base and did not include the memory bandwidth measurement tools added by the Memory Compression AHG. It is our understanding that MediaTek later merged in their code with the Memory Compression AHG version of HM 3.0.

# Analysis of Proposed Algorithms

## Version 1

Version 1 of JLCA allocates 120 bits to each 4×4 chroma block and gives 132 bits to each luma block. Therefore the compression algorithm is slightly different between luma and chroma.

### Chroma Compression

As described in [1], the compression for chroma is simply adaptive scaling with quantization of the minimum pixel value into 9 bits. The storage format is as follows: 2 bits for scale factor S, 9 bits for minimum pixel value, 4 bits to indicate position of the minimum pixel, and 7×15 bits to store the remaining 7-bit residuals. A fixed offset of (1 << (S-1)) is added to the reconstructed pixels when S > 0.

Inspection of the software confirms this method is implemented as described.

### Luma Compression

For luma, the scale factor is computed as the smallest S such that (|max - min| >> S) < 128. The scale factor S is stored using 2 bits. When S < 3, the minimum value is stored using 10 bits, the minimum position is stored using 4 bits, 11 residual values are stored using 8 bits, and 4 residuals are stored using 7 bits. When S=3, all pixels are quantized and represented using 8 bits.

The document [1] does not specify which residuals are stored using 7 bits when S<3. However, the accompanying presentation does. The positions of the pixels that are stored using 7-bit residuals are the 4-connected neighbors of the minimum pixel, with wrapping around the block borders.

Fixed reconstruction offsets are added when S<3.

Inspection of the software confirms this method is implemented as described in the presentation. The document should be revised to include the extra details in the presentation.

## Version 2

Version 2 of JLCA allocates a variable number of bits to the chroma blocks depending upon the scale factor computed for the block. The scale factor is computed as the smallest S such that (|max - min|>>S) < 64. Different coding schemes are used depending upon the value of S. The description in the document is clear for the cases of S<3. However, when S>=3, all that is said is “the Chroma block is compressed as in version 1 method.” Supposedly, the actual number of bits used to code each pair of co-located Cr and Cb blocks is used to compute the bit budget for the co-located luma blocks.

The description of the luma compression is unclear. There is no specification of the exact computations involved. The presentation is similarly obfuscated on this topic. The source code contains lots of if-then-else clauses to determine how many bits to allocate to each luma residual depending upon the chroma bit usage.

Even if the algorithm description were clarified, there are several complexity drawbacks to JLCA version 2. The processing of luma cannot proceed in parallel with chroma. The irregular processing of luma is also a concern.

## Version 1 with Luma Offset

There is little description of Version 1 with Luma Offset in the document. The presentation contains some more detail, but with an error. The luma storage format for S=3 shows 2 bits for S, S bits for offset, and 128 bits for the residuals, which totals 133 bits, not 132 as indicated in the presentation.

We did not receive the software that implements this scheme and cannot resolve the discrepancy noted above.

# Simulation Results

MediaTek shared with us an early version of their software that supports Version 1 and Version 2 of their proposed algorithms. This software used HM 3.0 with bug fix 146 as a base and did not include the memory bandwidth measurement tools added by the Memory Compression AHG.

## Version 1

We ran simulations using JLCA Version 1 software and produced the same RD data as MediaTek. However, we used HM 3.0 with modifications released by the Memory Compression AHG as reference. Therefore there are small differences in the BD-rate data because of the different anchors used. Also, we ran our anchor and JLCA on totally different clusters so the run times are meaningless.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table : MediaTek BD-Rate results for JLCA Version 1   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Y | U | V | Y | U | V | | Class A |  |  |  | 0.05 | 0.16 | 0.08 | | Class B | 0.36 | 0.18 | 0.19 | 0.07 | 0.01 | 0.04 | | Class C | 0.24 | 0.43 | 0.65 | 0.06 | 0.12 | 0.14 | | Class D | 0.07 | 0.01 | 0.83 | 0.09 | 0.15 | 0.14 | | Class E | 2.75 | 0.98 | 0.76 |  |  |  | | **Overall** | **0.70** | **0.35** | **0.57** | **0.07** | **0.10** | **0.10** | | Enc Time[%] | 100% | | | 99% | | | | Dec Time[%] | 106% | | | 105% | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table : Zenverge BD-Rate results for JLCA Version 1   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | Low delay B HE | | | Random Access HE | | | | Y | U | V | Y | U | V | | Class A |  |  |  | 0.04 | 0.01 | 0.05 | | Class B | 0.36 | 0.36 | 0.50 | 0.06 | -0.02 | 0.10 | | Class C | 0.21 | 0.37 | 0.62 | 0.06 | 0.13 | 0.09 | | Class D | 0.11 | 0.19 | 0.67 | 0.05 | 0.05 | 0.04 | | Class E | 2.79 | 0.80 | 0.65 |  |  |  | | **Overall** | **0.71** | **0.40** | **0.60** | **0.05** | **0.04** | **0.07** | | Enc Time[%] | 95% | | | 93% | | | | Dec Time[%] | 53% | | | 54% | | | |

## Version 2

We ran simulations using JLCA Version 2 software and produced the same RD data as MediaTek. However, we used HM 3.0 with modifications released by the Memory Compression AHG as reference. Therefore there are very small differences in the BD-rate data because of the different anchors used. Also, we ran our anchor and JLCA on totally different clusters so the run times are meaningless.

Table : MediaTek BD-Rate results for JLCA Version 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Low delay B HE | | | Random Access HE | | |
| Y | U | V | Y | U | V |
| Class A |  |  |  | 0.01 | 0.10 | 0.27 |
| Class B | 0.19 | 0.07 | 0.13 | 0.01 | -0.07 | -0.01 |
| Class C | 0.14 | 0.33 | 0.47 | 0.02 | 0.07 | 0.21 |
| Class D | -0.04 | 0.08 | 1.06 | 0.01 | -0.05 | 0.19 |
| Class E | 1.06 | 0.59 | 0.17 |  |  |  |
| **Overall** | **0.29** | **0.23** | **0.45** | **0.01** | **0.01** | **0.16** |
| Enc Time[%] | 100% | | | 99% | | |
| Dec Time[%] | 106% | | | 105% | | |
|  |

Table : Zenverge BD-Rate results for JLCA Version 2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Low delay B HE | | | Random Access HE | | |
| Y | U | V | Y | U | V |
| Class A |  |  |  | -0.01 | -0.05 | 0.24 |
| Class B | 0.19 | 0.25 | 0.44 | 0.01 | -0.10 | 0.05 |
| Class C | 0.12 | 0.27 | 0.45 | 0.02 | 0.08 | 0.16 |
| Class D | 0.00 | 0.27 | 0.89 | -0.03 | -0.15 | 0.09 |
| Class E | 1.10 | 0.39 | 0.05 |  |  |  |
| **Overall** | **0.30** | **0.29** | **0.48** | **0.00** | **-0.06** | **0.13** |
| Enc Time[%] | 95% | | | 93% | | |
| Dec Time[%] | 52% | | | 52% | | |

# Final Comments

Given that the details of sharing bits between luma and chroma blocks depend upon the relative number of co-located blocks, it is clear that application of JLCA to YUV422 and YUV444 formats will involve a redesign for each case. Similarly, extending the JLCA to handle N-to-8 for values of N other than 10 will also involve extensive redesign for each different value of N.

Since each luma block will need to access bits outside a 128-bit chunk, there are some complications that need to be addressed in the memory bandwidth calculation routines used by the Memory Compression AHG. A new memory organization scheme needs to be considered for JLCA. One simple approach is to ignore any byte alignment issues and specify the actual compression ratio for luma that is different from that for chroma. For example, for Version 1, the luma compression ratio can be expressed as 160:132 and the chroma compression ratio as 160:120.

For Version 2, the number of bits allocated to a luma block depends upon the number of bits actually used in the co-located chroma blocks. Therefore a memory access to a luma region would necessitate a memory access to a chroma block. Most of the time, an access to a luma block would also require an access to the co-located chroma blocks. In cases where there is no need to access chroma at the same time as luma (such as due to different interpolation filter support regions), Version 2 would incur memory bandwidth and computational overhead. This aspect needs to be studied more carefully.

# References

1. S. Liu, X. Zhang, and S. Lei, “Joing luma-chroma adaptive reference picture memory compression,” JCT-VC Document JCTVC-F073, Torino, July 2011.
2. T. Chujoh and T. Yamakage, “Adaptive scaling for reference pictures memory compression,” JCT-VC Document JCTVC-E133, Geneva, March 2011.
3. D. Hoang, “Unified scaling with adaptive offset for reference frame compression with IBDI,” JCT-VC Document JCTVC-D035, Daegu, January 2011.
4. D. Hoang, “Unified scaling with adaptive offset for reference frame compression,” JCT-VC Document JCTVC-F075, 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).**