|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, 21-30 Nov, 2011 | Document: JCTVC-G154  WG11 Number: |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Title:* | **De-quantization without Rounding Offset** | | | |
| *Status:* | Input Document to JCT-VC | | | |
| *Purpose:* | Proposal | | | |
| *Author(s) or Contact(s):* | Ximin Zhang  Shan Liu Shawmin Lei  2860 Junction Ave.  San Jose, CA95134, USA | *Emails:* | ximin.zhang@mediatek.com shan.liu@mediatek.com shawmin.lei@mediatek.com |
| *Source:* | MediaTek USA Inc. | | | |

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

# Abstract

This contribution proposes to remove the round offset from dequantization process. By doing this, one addition operation can be saved for each coefficient and in many cases, one shift operation can be saved for each coefficient as well. Experimental results report identical BD-rates under common test condition. Very small bit impacts on bit rate and PSNR (up to 0.x% bit rate increase and/or up to 0.04dB PSNR degradation at >50dB PSNR quality) are reported for some very high bit rate (small QP) cases.

1. Introduction

Various Transforms have been widely used in image and video compression. Among them, discrete cosine transform (DCT) has been adopted in the video coding standards such as MPEG2 and H.264/AVC due to its energy compaction property. In H.264/AVC, a scaled approximation of DCT transform is adopted such that the transform can be conducted by simple arithmetical computation. During the coding process, the transform and quantization computation are combined and restructured into a cascaded core transform plus scaling plus shifting process. Correspondingly, the dequantization is also conducted by scaling plus shifting process in the decoder. In doing so, the computational complexity is significantly reduced.

In HEVC, similar approaches have been adopted for quantization and dequantization. The dequantization process is in HM4.0 is described as follows.

UInt iTransformShift = MAX\_TR\_DYNAMIC\_RANGE - uiBitDepth - uiLog2TrSize;

iShift = QUANT\_IQUANT\_SHIFT - QUANT\_SHIFT - iTransformShift;

iAdd = 1 << (iShift-1);

scale = g\_invQuantScales[m\_cQP.m\_iRem] << m\_cQP.m\_iPer;

for( Int n = 0; n < iWidth\*iHeight; n++ )

{

iCoeffQ = ((piQCoef[n]\*scale +iAdd)>>iShift; (1)

piCoef[n] = Clip3(-32768,32767,iCoeffQ);

}

The constants utilized in the above process are defined as following.

#define QUANT\_IQUANT\_SHIFT 20 // Q(QP%6) \* IQ(QP%6) = 2^20

#define QUANT\_SHIFT 14 // Q(4) = 2^14

#define MAX\_TR\_DYNAMIC\_RANGE 15 // Maximum transform dynamic range

UInt g\_invQuantScales[6] = {40,45,51,57,64,72};

The corresponding parameters are calculated as follows.

m\_iPer = (iQP + 6\*g\_uiBitIncrement)/6; (2)

m\_iRem = (iQP + 6\*g\_uiBitIncrement)%6;

g\_uiBitIncrement = m\_uiInternalBitDepth - g\_uiBitDepth;

uiBitDepth = g\_uiBitDepth + g\_uiBitIncrement;

UInt g\_uiBitDepth = 8; // base bit-depth

uiLog2TrSize is previously calculated by log2(TrSize) where TrSize=4,8,16,32. In current HM common test condition.

Based on the above calculation, the shifting parameter “iShift” can be calculated as following

iShift = QUANT\_IQUANT\_SHIFT - QUANT\_SHIFT - MAX\_TR\_DYNAMIC\_RANGE + uiBitDepth + uiLog2TrSize

= uiBitDepth+ uiLog2TrSize-9

= g\_uiBitDepth + g\_uiBitIncrement + log2(TrSize)-9

= g\_uiBitIncrement + log2(TrSize)-1 //when g\_uiBitDepth = 8. (3)

# Proposed Method

In this contribution, we propose to remove the rounding offset, i.e. “iAdd” from the dequantization process, as shown in the following Equation 4.

iCoeffQ = (piQCoef[n]\*scale>>iShift; (4)

Equation (1) is equivalent to equation (4) if one of the following two conditions is met.

**Condition 1:** m\_cQP.m\_iPer >= iShift

**Condition 2:** (piQCoef[n]\*scale)% 2iShift =0

Based on equation (2) and (3), m\_iPer = iQP/6 + g\_uiBitIncrement, iShift = g\_uiBitIncrement + log2(TrSize)-1. Therefore, condition 1 is equivalent to

iQP/6>= log2(TrSize)-1. (5)

In current HM common test condition, the maximum value of “TrSize” is 32. Hence, as long as iQP/6>=4, i.e. iQP >=24, equation (5) is satisfied regardless of the transform size that is used. That is, the shifting offset “iAdd” is useless and thus this addition operation is wasted when “iQP>=24”. In the cases when “iQP<24”, if TrSize<=16, equation (5) is satisfied as long as iQP>=18; if TrSize<=8, equation (5) is satisfied as long as iQP>=12; so on and so forth.

When “m\_cQP.m\_iPer < iShift”, condition 2 can be rewritten as follows,

(piQCoef[n]\*g\_invQuantScales[])% 2iShift-m\_cQP.m\_iPer =0 (6)

and

(piQCoef[n]\*g\_invQuantScales[])% 2 (log2(TrSize)-1 – iQP/6) =0 (7)

Since the maximum of “log2(TrSize)” is 5 (under the current HM common test condition), the value of “(log2(TrSize)-1 – iQP/6)” can be {4, 3, 2, 1}. Correspondingly, equation (6) and (7) may be equivalent to (piQCoef[n]\*g\_invQuantScales[])% 16=0, (piQCoef[n]\*g\_invQuantScales[])% 8=0, (piQCoef[n]\*g\_invQuantScales[])% 4=0 and (piQCoef[n]\*g\_invQuantScales[])% 2=0.

Based on above analyses, it can be mathematically proved that the proposed method would result in identical BD-rates as HM4.0 anchor under the common test condition, which is shown in Table 1. The bit rates and/or PSNR might be a little different by using the proposed method compared with HM4.0 anchor in some very small QP cases. Worst case results are shown in Table 2.

The proposed implementation is as follows.

if(m\_cQP.m\_iPer >= iShift)

 {

     scale = g\_invQuantScales[m\_cQP.m\_iRem] << (m\_cQP.m\_iPer-iShift);

     for( Int n = 0; n < iWidth\*iHeight; n++ ) {

        iCoeffQ =  piQCoef[n] \* scale;

        piCoef[n] = Clip3(-32768,32767,iCoeffQ);

     }

}

else

 {

      m\_shift = iShift - m\_cQP.m\_iPer;

      scale = g\_invQuantScales[m\_cQP.m\_iRem];

      for( Int n = 0; n < iWidth\*iHeight; n++ ) {

        iCoeffQ =  (piQCoef[n] \* scale) >> m\_shift;

        piCoef[n] = Clip3(-32768,32767,iCoeffQ);

      }

}

Therefore, one addition and one shift operation can be saved for each coefficient when m\_cQP.m\_iPer >= iShift; or, one addition operation can be saved for each coefficient otherwise.

# Experimental Results

Simulations were conducted following common test conditions defined in JCTVC-F900 [1]. Anchor data was generated using HM4.0 software [2]. Table 1 reports the results under common test condition. As analyzed in the previous section, we see all identical BD-rates between the proposed method and the anchor.

Table 1 Results for the proposed dequantization without rounding offset, compared with HM4.0 anchor under common test condition.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **All Intra HE** | | | **All Intra LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class B | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class C | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class D | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class E | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| **Overall** | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
|  | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Enc Time[%] | 100% | | | 99% | | |
| Dec Time[%] | 99% | | | 99% | | |
|  |  |  |  |  |  |  |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class B | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class C | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class D | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class E |  |  |  |  |  |  |
| **Overall** | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
|  | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 100% | | | 99% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class C | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class D | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class E | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| **Overall** | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
|  | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 100% | | | 100% | | |
|  |  |  |  |  |  |  |
|  | **Low delay P HE** | | | **Low delay P LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class C | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class D | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Class E | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| **Overall** | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
|  | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 100% | | | 100% | | |

Table 2 reports the “worse case” results. We tested some small QP cases in which the proposed method may have higher impact on bit rate and PSNR. Among all test cases (all 20 test sequences, all eight configurations and various small QPs) we picked a couple of worst cases to report below. Both happened in All Intra HE setting.

Table 2 Worst-case results for the proposed dequantization without rounding offset.

1. “SteamLocomotive”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **QP = 9** | **Bit rate (kbps)** | **Y PSNR (dB)** | **U PSNR (dB)** | **V PSNR (dB)** |
| HM4.0 | 575965.07 | 54.13 | 53.72 | 53.58 |
| Proposed | 580236.65 | 54.09 | 53.68 | 53.54 |
| Difference | +0.7% | -0.04 | -0.04 | -0.04 |

1. “Kimono”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **QP = 9** | **Bit rate (kbps)** | **Y PSNR (dB)** | **U PSNR (dB)** | **V PSNR (dB)** |
| HM4.0 | 156313.10 | 53.80 | 53.75 | 53.90 |
| Proposed | 157262.59 | 53.80 | 53.74 | 53.91 |
| Difference | +0.6% | -0.00 | -0.01 | +0.01 |

# Conclusions

This contribution reports the method and results for removing round offset from dequantization process. By doing this, one addition operation can be saved for each coefficient and in many cases, one shift operation can be saved for each coefficient as well. Experimental results report identical BD-rates under common test condition. Very small bit impacts on bit rate and PSNR (up to 0.x% bit rate increase and/or up to 0.04dB PSNR degradation at >50dB PSNR quality) are reported for some very high bit rate (small QP) cases. It is recommended to adopt the proposed method in HM.

# References

1. Frank Bossen, “Common test conditions and software reference configurations”, JCTVC-F900, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG, Torino, Italy, July 2011.
2. HM 4.0 Software, <http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-4.0>.

# Patent rights declaration(s)

**MediaTek 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).**