|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  7th Meeting: Geneva, CH, 21-30 November, 2011 | Document: JCTVC-G780 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | **CE3: The worst case memory band-width reduction by 2D->1D interpolation replacement (from Samsung).** | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Elena Alshina Alexander Alshin  Jianle Chen  JeongHoon Park  #416 Matean 3-dong, Yeongtong-gu, Suwon-si, Gyeonggi-do, 443-742, Korea | Email: | elena\_a.alshina@samsung.com alexander\_b.alshin@samsung.com  [jianle.chen@samsung.com](mailto:jianle.chen@samsung.com)  jeonghoon@samsung.com |
| *Source:* | Samsung Electronics Co., Ltd. | | |

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

# Abstract

High memory access during motion compensation is the main drawback for both bi-prediction and long-taps interpolation filters. In this contribution motion compensation process in bi-prediction is modified to reduce memory access: a motion vector which requires high memory access is replaced by nearest one with acceptable level for memory access.

# Introduction

High memory access during motion compensation is the main drawback for both bi-prediction and long-taps interpolation filters. In this contribution following motion compensation process modification is proposed. Memory access depends on prediction direction (bi-/uni-prediction), filter length, block size and fractional position of motion vector. If both components of MV are fractional then interpolation will have to stages: vertical and horizontal (let’s call it 2D-interpoaltion). If one component of MV is integer then only 1D interpolation is required. In this case memory access is noticeable smaller than for 2D-int5erpoaltion. If both components of MV are integer then no interpolation is needed and this is simplest case both from view point of complexity and memory access.

All cases can be classified in decreasing order of memory access. Such kind of analysis was done in CE3; results are included into CE3 template. For different memory architectures absolute numbers for memory band-width are different but the worst case in HM4.0 is always 4x8 (WxH) Luma bi-predictive blocks with 2D-2D interpolation (2D interpolation is needed for both references). The second rank is 4x8 (WxH) Luma 2D-2D bi-predictive blocks and so on.

For cases when memory access exceeds predefined limit we propose to replace MV with nearest one which will keep memory access under predefined limit. An example of 2D🡪1D interpolation replacement is shown on Fig 1.

|  |
| --- |
|  |
| Fig.1 MV modification for 2D🡪1D interpolation replacement. |

Here we present several level for the worst case memory band-width reduction.

# Test results

We implemented Mv modification for memory band-width reduction in HM4.0 and tested according to CE3 test description [1] under common test conditions[2]. Attached excel spread-sheep provide detail information of our tests.

We would like to thank Sony for comprehensive cross-check of these tests.

## Level 1

In this test we avoid 2D-2D interpolation in 4x8 and 8x4 Luma and 2x4 and 4x2 Chroma bi-predictive blocks. But 2D-1D interpolation is not restricted: 2D interpolation is possible in one reference only.

Performance results are presented in Table 1.

**Table 1**. Performance test results if 2D-2D interpolation in 4x8 and 8x4 Luma and 2x4 and 4x2 Chroma bi-predictive blocks is replaced by 2D-1D.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.0% | 0.0% | -0.1% | 0.0% | 0.2% | 0.1% |
| Class B | 0.0% | 0.1% | 0.1% | 0.0% | 0.1% | 0.1% |
| Class C | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.1% |
| Class D | 0.1% | 0.2% | 0.1% | 0.1% | 0.2% | 0.1% |
| Class E |  |  |  |  |  |  |
| **Overall** | 0.1% | 0.1% | 0.0% | 0.1% | 0.1% | 0.1% |
|  | 0.1% | 0.1% | 0.0% | 0.0% | 0.1% | 0.1% |
| Enc Time[%] | 100% | | | 101% | | |
| Dec Time[%] | 100% | | | 102% | | |
| Mult[%] | 0% | | | 0% | | |
| Adds[%] | 0% | | | 0% | | |
| Access[%] | -1% | | | -1% | | |
| MemBand(2D) Ave.[%] | 0% | | | 0% | | |
| MemBand(2D) Max.[%] | 5% | | | 4% | | |
| MemBand(1D) Ave.[%] | 0% | | | 0% | | |
| MemBand(1D) Max.[%] | 4% | | | 6% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.0% | 0.2% | -0.1% | 0.0% | 0.1% | 0.0% |
| Class C | 0.1% | 0.1% | 0.2% | 0.0% | 0.2% | -0.1% |
| Class D | 0.0% | 0.6% | 0.1% | 0.1% | 0.2% | 0.2% |
| Class E | 0.0% | 0.3% | 0.6% | -0.1% | 0.3% | 0.1% |
| **Overall** | 0.0% | 0.3% | 0.1% | 0.0% | 0.2% | 0.1% |
|  | 0.0% | 0.3% | 0.1% | 0.0% | 0.2% | 0.1% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 101% | | | 102% | | |
| Mult[%] | -1% | | | 0% | | |
| Adds[%] | -1% | | | 0% | | |
| Access[%] | -1% | | | -1% | | |
| MemBand(2D) Ave.[%] | 0% | | | 0% | | |
| MemBand(2D) Max.[%] | 3% | | | 9% | | |
| MemBand(1D) Ave.[%] | -56% | | | 0% | | |
| MemBand(1D) Max.[%] | 3% | | | 9% | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Gain Ave.[%] | **0.03%** | **0.11%** | **0.05%** |
| MemBand(2D) Ave.[%] | -0.09% | | |
| MemBand(2D) Worst[%] | -12.5% | | |
| MemBand(1D) Ave.[%] | -9.44% | | |
| MemBand(1D) Worst[%] | -15.87% | | |

## Level 2

In this test we restrict any 2D in 4x8 and 8x4 Luma and 2x4 and 4x2 Chroma bi-predictive blocks. The worst case for memory band-width is Luma 4x8 and Chroma 2x4 bi-predictive blocks with 1D-1D interpolation.

Performance results are presented in Table 2.

**Table 2**. Performance test results if no 2D interpolation in 4x8 and 8x4 Luma and 2x4 and 4x2 Chroma bi-predictive blocks is performed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.0% | -0.2% | -0.1% | 0.0% | 0.1% | 0.0% |
| Class B | 0.0% | 0.1% | 0.1% | 0.0% | 0.0% | 0.2% |
| Class C | 0.1% | 0.1% | 0.1% | 0.1% | 0.1% | 0.2% |
| Class D | 0.2% | 0.3% | 0.2% | 0.2% | 0.1% | 0.2% |
| Class E |  |  |  |  |  |  |
| **Overall** | 0.1% | 0.1% | 0.0% | 0.1% | 0.1% | 0.1% |
|  | 0.1% | 0.1% | 0.0% | 0.1% | 0.1% | 0.1% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 101% | | | 101% | | |
| Mult[%] | 0% | | | -1% | | |
| Adds[%] | 0% | | | -1% | | |
| Access[%] | -1% | | | -1% | | |
| MemBand(2D) Ave.[%] | 0% | | | -1% | | |
| MemBand(2D) Max.[%] | 0% | | | -1% | | |
| MemBand(1D) Ave.[%] | -1% | | | -1% | | |
| MemBand(1D) Max.[%] | 0% | | | 0% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.0% | 0.0% | 0.1% | 0.1% | 0.2% | 0.2% |
| Class C | 0.1% | 0.0% | 0.0% | 0.1% | 0.1% | 0.0% |
| Class D | 0.2% | 0.8% | 0.6% | 0.2% | 0.4% | 0.0% |
| Class E | 0.0% | 0.0% | 0.1% | 0.0% | 0.3% | 0.3% |
| **Overall** | 0.1% | 0.2% | 0.2% | 0.1% | 0.3% | 0.1% |
|  | 0.1% | 0.2% | 0.2% | 0.1% | 0.2% | 0.1% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 101% | | | 103% | | |
| Mult[%] | -1% | | | -1% | | |
| Adds[%] | -1% | | | -1% | | |
| Access[%] | -1% | | | -1% | | |
| MemBand(2D) Ave.[%] | 0% | | | 0% | | |
| MemBand(2D) Max.[%] | 9% | | | 4% | | |
| MemBand(1D) Ave.[%] | 0% | | | 0% | | |
| MemBand(1D) Max.[%] | 9% | | | 5% | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Gain Ave.[%] | 0.05% | 0.10% | 0.08% |
| MemBand(2D) Worst[%] | -25% | | |
| MemBand(1D) Worst[%] | -31% | | |

## Level 3

In this test additionally to Level 2 we restricted MV Luma 4x8 and Chroma 2x4 bi-predictive blocks to be integer only.

Performance results are presented in Table 3.

**Table 3**. Performance test results if no 2D interpolation in 8x4 Luma and 4x2 Chroma bi-predictive blocks is performed and MV in Luma 4x8 and Chroma 2x4 bi-predictive blocks is integer

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Random Access HE** | | | **Random Access LC** | | |
|  | Y | U | V | Y | U | V |
| Class A | 0.0% | -0.1% | -0.1% | 0.0% | 0.2% | 0.1% |
| Class B | 0.1% | 0.1% | 0.1% | 0.0% | 0.1% | 0.2% |
| Class C | 0.1% | 0.1% | 0.3% | 0.2% | 0.2% | 0.2% |
| Class D | 0.3% | 0.5% | 0.4% | 0.3% | 0.4% | 0.4% |
| Class E |  |  |  |  |  |  |
| **Overall** | 0.1% | 0.2% | 0.2% | 0.1% | 0.2% | 0.2% |
|  | 0.1% | 0.2% | 0.2% | 0.1% | 0.2% | 0.2% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 100% | | | 101% | | |
| Mult[%] | -1% | | | -1% | | |
| Adds[%] | -1% | | | -1% | | |
| Access[%] | -1% | | | -1% | | |
| MemBand(2D) Ave.[%] | 0% | | | 0% | | |
| MemBand(2D) Max.[%] | 4% | | | 7% | | |
| MemBand(1D) Ave.[%] | 0% | | | 0% | | |
| MemBand(1D) Max.[%] | 4% | | | 7% | | |
|  |  |  |  |  |  |  |
|  | **Low delay B HE** | | | **Low delay B LC** | | |
|  | Y | U | V | Y | U | V |
| Class A |  |  |  |  |  |  |
| Class B | 0.1% | 0.1% | 0.4% | 0.1% | 0.1% | 0.1% |
| Class C | 0.2% | 0.1% | 0.1% | 0.2% | 0.4% | 0.2% |
| Class D | 0.4% | 0.7% | 0.9% | 0.4% | 0.5% | 0.6% |
| Class E | 0.0% | 0.0% | 0.5% | 0.0% | 0.0% | 0.3% |
| **Overall** | 0.2% | 0.2% | 0.5% | 0.2% | 0.3% | 0.3% |
|  | 0.2% | 0.2% | 0.4% | 0.2% | 0.3% | 0.3% |
| Enc Time[%] | 100% | | | 100% | | |
| Dec Time[%] | 101% | | | 102% | | |
| Mult[%] | -1% | | | -1% | | |
| Adds[%] | -1% | | | -1% | | |
| Access[%] | -1% | | | -1% | | |
| MemBand(2D) Ave.[%] | -1% | | | -1% | | |
| MemBand(2D) Max.[%] | 6% | | | 7% | | |
| MemBand(1D) Ave.[%] | -1% | | | -1% | | |
| MemBand(1D) Max.[%] | 6% | | | 8% | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Gain Ave.[%] | 0.10% | 0.15% | 0.20% |
| MemBand(2D) Worst[%] | -36% | | |
| MemBand(1D) Worst[%] | -31% | | |

# Summary of results and suggestions

Table 4 presents the summary of test results.

Table 4. Summary of 3 tests.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Bi-prediction in | BD-rate Y | BD-rate U | BD-rate V | MemBand(2D) Worst[%] | MemBand(1D) Worst[%] |
| Luma 8x4, Chroma 4x2  Luma 4x8, Chroma 2x4 |
| 1 | Both: 2D-2D🡪2D-1D | 0.03% | 0.11% | 0.05% | -12% | -16% |
| 2 | Both: 2D-1D,2D-2D🡪1D-1D | 0.05% | 0.10% | 0.08% | -25% | -31% |
| 3 | 2D-1D,2D-2D🡪1D-1D  only integer | 0.10% | 0.15% | 0.20% | -36% | -31% |

Based on test results above Level 2 of MV restriction seems to be good trade-off. With minimal losses in performance the worst case of memory band-width was reduced by at least 25% (depends on memory architecture).

We would like to notice that proposed change can be combined with almost any interpolation filters change proposed and tested in CE3.

# Proposed WD changes

Proposed changes for WD are highlighted in yellow below.

##### Fractional sample interpolation process

Inputs to this process are:

– a luma location ( xC, yC ) specifying the top-left luma sample of the current coding unit relative to the top left luma sample of the current picture,

– a luma location ( xB, yB ) specifying the top-left luma sample of the current prediction unit relative to the top left luma sample of the current coding unit,

– the width and height of this prediction unit, nPSW and nPSH, in luma-sample units,

– a luma motion vector mvLX given in quarter-luma-sample units,

– a chroma motion vector mvCLX given in eighth-chroma-sample units,

– the selected reference picture sample arrays refPicLXL, refPicLXCb, and refPicLXCr.

Outputs of this process are:

– a (nPSW)x(nPSH) array predSampleLXL of prediction luma sample values,

– two (nPSW/2)x(nPSH/2) arrays predSampleLXCb, and predSampleLXCr of prediction chroma sample values.

The location ( xP, yP ) given in full-sample units of the upper-left luma samples of the current prediction unit relative to the upper-left luma sample location of the given reference sample arrays is derived by

xP = xC + xB (8‑87)  
yP = yC + yB (8‑88)

Let ( xIntL, yIntL ) be a luma location given in full-sample units and ( xFracL, yFracL ) be an offset given in quarter-sample units. These variables are used only inside this subclause for specifying general fractional-sample locations inside the reference sample arrays refPicLXL, refPicLXCb, and refPicLXCr.

If bi-prediction is performed and block size is smaller than 8x8 Luna and smaller than 4x4 for Chroma then MV regulation process is needed.

{

Motion vector regulation.

iMvAccu = 8 for Luma and iMvAccu = 4 for Chroma;

iMvAccuShift = 3 for Luma and iMvAccu = 2 for Chroma;

iMvAccuMinus1 = iMvAccu - 1 ;

xFrac = rcMv.getHor() & iMvAccuMinus1;

yFrac = rcMv.getVer() & iMvAccuMinus1;

if(xFrac && yFrac)

{

xFracDist = min (xFrac, iMvAccu - xFrac);

yFracDist = min (yFrac, iMvAccu - yFrac);

if(xFracDist < yFracDist)

{

if (rcMv.getHor() > 0)

{

iMvX = ((rcMv.getHor() + (iMvAccu >> 1) - 1) >> iMvAccuShift) << iMvAccuShift;

}

else

{

iMvX = ((rcMv.getHor() + (iMvAccu >> 1)) >> iMvAccuShift) << iMvAccuShift;

}

iMvY = rcMv.getVer();

}

else

{

iMvX = rcMv.getHor();

if (rcMv.getVer() > 0)

{

iMvY = ((rcMv.getVer() + (iMvAccu >> 1) - 1) >> iMvAccuShift) << iMvAccuShift;

}

else

{

iMvY = ((rcMv.getVer() + (iMvAccu >> 1)) >> iMvAccuShift) << iMvAccuShift;

}

}

rcMv.set(iMvX, iMvY);

}

}

For each luma sample location ( 0 <= xL < nPSW, 0 <= yL < nPSH ) inside the prediction luma sample array predSampleLXL, the corresponding prediction luma sample value predSampleLXL[xL, yL] is derived as follows:

– The variables xIntL, yIntL, xFracL, and yFracL are derived by

xIntL = xP + ( mvLX[ 0 ] >> 2 ) + xL (8‑89)  
yIntL = yP + ( mvLX[ 1 ] >> 2 ) + yL (8‑90)

xFracL = mvLX[ 0 ] & 3 (8‑91)  
yFracL = mvLX[ 1 ] & 3 (8‑92)

– The prediction luma sample value predSampleLXL[ xL, yL ] is derived by invoking the process specified in subclause 8.4.2.2.2.1 with ( xIntL, yIntL ), ( xFracL, yFracL ) and refPicLXL given as input.

Let ( xIntC, yIntC ) be a chroma location given in full-sample units and ( xFracC, yFracC ) be an offset given in one-eighth sample units. These variables are used only inside this subclause for specifying general fractional-sample locations inside the reference sample arrays refPicLXCb and refPicLXCr.

For each chroma sample location ( 0 <= xC < nPSW/2, 0 <= yC < nPSH/2) inside the prediction chroma sample arrays predSampleLXCb and predSampleLXCr, the corresponding prediction chroma sample values predSampleLXCb[ xC, yC ] and predSampleLXCr[ xC, yC ] are derived as follows:

– The variables xIntC, yIntC, xFracC, and yFracC are derived by

xIntC = ( xP / 2 ) + ( mvCLX[ 0 ] >> 3 ) + xC (8‑93)  
yIntC = ( yP / 2 ) + ( mvCLX[ 1 ] >> 3 ) + yC (8‑94)

xFracC = mvLX[ 0 ] & 7 (8‑95)  
yFracC = mvLX[ 1 ] & 7 (8‑96)

– The prediction sample value predSampleLXCb[ xC, yC ] is derived by invoking the process specified in subclause 8.4.2.2.2.2 with ( xIntC, yIntC ), ( xFracC, yFracC ) and refPicLXCb given as input.

– The prediction sample value predSampleLXCr[ xC, yC ] is derived by invoking the process specified in subclause 8.4.2.2.2.2 with ( xIntC, yIntC ), ( xFracC, yFracC ) and refPicLXCr given as input.

# Conclusions

Based on test results

* At least 25% the worst case memory band-width reduction
* With 0.05% (Y), 0.10%(U), 0.08% (V) average losses

we would like to propose to modify MV in smallest bi-predictive blocks (4x8, 8x4 Luma and 4x2,2x4 Chroma) as it is show on Fig 1 and avoid 2D interpolation in these blocks.

# References

1. JCT-VC, “CE3: Motion Compensation”, JCTVC-F903, 6th JCT-VC Meeting, Torino, Italy, 14-22 July, 2011
2. JCT-VC, “Common test conditions and software reference configurations”, JCTVC-F900, 6th JCT-VC Meeting, Torino, Italy, 14-22 July, 2011

# Patent rights declaration(s)

**Samsung Electronics Co., Ltd. may have current or pending patent rights 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).**