A Resolution-adaptive Interpolation Filter for Video Codec

Hao Lv[1,2], Ronggang Wang[1*], Yuan Li[2], Chuang Zhu[2], Huizhu Jia[2], Xiaodong Xie[2] and Wen Gao[2]
E-mail: hlv@pku.edu.cn, rgwang@pkusz.edu.cn, {yuanli, czhu, hzjia, donxie, wgao}@pku.edu.cn
[1]School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen, China
[2]National Engineering Laboratory of Video Technology, Peking University, Beijing, China

Abstract—**The fraction-pel interpolation filter varies in the video coding standards such as H.264/AVC, AVS and HEVC. Since fractional-pel motion compensation plays an important role in the video encoder, the interpolation of fractional-pel pixels can be refined and designed better to enhance the coding efficiency. In this paper, we firstly propose the generation algorithm of interpolation filter coefficients, and four different tap filters, namely 4tap, 6 tap, 8 tap and 10tap, are tested. A resolution-adaptive interpolation filter for different resolution videos is then introduced based on this algorithm to achieve the maximum bitrate saving. In the proposed scheme, 4 tap filter is applied for the UHD (2560×1600 and above) videos, 6 tap filter and 10 tap filter are performed in the videos whose resolution ranging from 720P (1280×720) to 1080P (1920×1080) and the videos with the resolution below 720P, respectively. When 4 tap filter and 6 tap filter are used in high-definition video, the coding efficiency can increase and the computational complexity will reduce greatly, which is actually beneficial to make hardware optimization more effectively especially SIMD (Single Instruction Multiple Data) and VLSI design. Experiments show that the average BD-rate gains on luma Y, chroma U and V are 1.4%, 0.7% and 0.7% for LP-Main configuration, when conducted in HEVC reference software HM11.0. The coding efficiency gains are significant for some video sequences and can reach up to 6.1%.**

Keywords—Resolution-adaptive; HEVC; interpolation

I. INTRODUCTION

Motion-compensation prediction is the key technique for efficient video compression and enhance coding performance obviously. The fractional-pel motion search method is similar among the modern video compression standards such as H.264/AVC, AVS and HEVC, however the fractional-pel interpolation methods vary according to different standards. H.264/AVC [1] uses the 6 tap FIR filter to perform half-pixel interpolation and the average filter to perform quarter-pixel interpolation further for luma components. In Chinese video coding standard AVS [2], a 4 tap filter is used for half-pel interpolation and horizontal/vertical quarter-pel interpolation, while the average filter is further used for the diagonal quarter-pel pixels. Since HM5.0, HEVC uses "7q(1/4)+8h" DCTIF (DCT-based interpolation filter) for 1/4 luma interpolation which means 7 tap DCTIF is used for quarter-pel positions and 8 tap DCTIF is used for half-pel positions [3, 4]. Considering the time-varying statistics of video sources, different kinds of interpolation filter are proposed. The adaptive interpolation filter (AIF) [5] technique optimizes and also codes the filter coefficients on a frame basis. Directional AIF (D-AIF) [6] can interpolate the fractional-pel pixels in a form of diagonal cross and many optimizations [7, 8] are also proposed to enhance the performance. Reference [9] proposes a parametric interpolation filter (PIF), which uses a function determined by five parameters instead of by individual coefficients to generate the filter. However, the computation complexity is obviously high. This paper is mainly to propose a resolution-adaptive interpolation filter to enhance the coding performance and reduce the complexity simultaneity for high-definition video.

The rest of the paper is organized in four sections. In Section 2, we will present the method of proposed fractional-pel interpolation filter in detail. Then a comprehensive analysis of the fractional-pel interpolation is given, including pixel access and computation complexity. In Section 3, the experimental results are provided. At last, we give a brief conclusion in Section 4.

II. RESOLUTION-ADAPTIVE INTERPOLATION SCHEME

A. *Proposed interpolation filter*

We proposed a new interpolation filter based on the Lanczos filter, which is 2D separable. Lanczos filter can be obtained by adding a Lanczos window to the function *sinc*, which is the function of ideal low pass filter in time domain. The interpolation filter coefficients calculated by (1) and (2) are real (not integer) numbers. Here x means the fractional position, n means the filter tap and generally n is odd. In practical application, filter coefficients are scaled by some factor (2^s is preferable) and rounded to integer, as described in (3). Here s represents the accuracy of filter coefficients. After scaling, the coefficients should keep the normalization condition as (4). According to (1) and (2), we can derive that the filter coefficients for fractional positions $x < 1/2$ and the coefficients for $x > 1/2$ are mirror symmetry.

$$L_n(x) = \begin{cases} \mathrm{sinc}(x)\mathrm{sinc}(\frac{x}{n}) & , -n < x < n \\ 1 & , x = 0 \\ 0 & , otherwise \end{cases} \tag{1}$$

$$\mathrm{sinc}(x) = \frac{\sin \pi x}{\pi x} \tag{2}$$

$$Filter_n(x) = round\left(L_n(x) \times 2^s\right) \tag{3}$$

$$\sum Filter_n(x) = 2^s \tag{4}$$

According to the above formulas, two main factors that influence the coefficients of interpolation filter are the accuracy of coefficients and the number of filter tap. Generally we set the accuracy factor s to 6, because when s is more than 6, say 8, there is no obvious performance gain in our experiment. Furthermore, since the filter input bit-depth of

---

*The corresponding author.

the video source can be 8 or 10, the intermediate buffer can be reduced to 16 bits and all data after each of the vertical and horizontal filtering passes will hold in 16-bit memory when coefficient accuracy is set to 6 [10]. For some application it is important to store intermediate data within 16 bits, especially when SIMD-like operation is used. Table I~IV show the optimal filter coefficients of different taps after optimization.

TABLE I. COEFFICIENTS OF 4 TAP FILTERS (S=6)

Position	Filter coefficients	Mults	Adds
1/4	{-6, 56, 15, -1}	3	3
2/4	{-4, 36, 36, -4}	4	3
3/4	{-1, 15, 56, -6}	3	3

TABLE II. COEFFICIENTS OF 6 TAP FILTERS (S=6)

Position	Filter coefficients	Mults	Adds
1/4	{2, -9, 57, 17, -4, 1}	5	5
2/4	{2, -9, 39, 39, -9, 2}	6	5
3/4	{1, -4, 17, 57, -9, 2}	5	5

TABLE III. COEFFICIENTS OF 8 TAP FILTERS (S=6)

Position	Filter coefficients	Mults	Adds
1/4	{-1, 4, -10, 57, 18, -6, 3, -1}	6	7
2/4	{-1, 4, -11, 40, 40, -11, 4, -1}	6	7
3/4	{-1, 3, -6, 18, 57, -10, 4, -1}	6	7

TABLE IV. COEFFICIENTS OF 10 TAP FILTERS (S=6)

Position	Filter coefficients	Mults	Adds
1/4	{1, -2, 4, -10, 57, 19, -7, 3, -1, 0}	7	8
2/4	{1, -2, 5, -12, 40, 40, -12, 5, -2, 1}	8	9
3/4	{0, -1, 3, -7, 19, 57, -10, 4, -2, 1}	7	8

$$a_{0,0} = \left(-6\times A_{-1,0} + 56\times A_{0,0} + 15\times A_{1,0} - A_{2,0}\right) \gg shift1 \tag{5a}$$

$$b_{0,0} = \left(-4\times A_{-1,0} + 36\times A_{0,0} + 36\times A_{1,0} - 4\times A_{2,0}\right) \gg shift1 \tag{5b}$$

$$c_{0,0} = \left(-A_{-1,0} + 15\times A_{0,0} + 56\times A_{1,0} - 6\times A_{2,0}\right) \gg shift1 \tag{5c}$$

$$e_{0,0} = \left(-6\times a'_{0,-1} + 56\times a'_{0,0} + 15\times a'_{0,1} - a'_{0,2}\right) \gg shift2 \tag{6a}$$

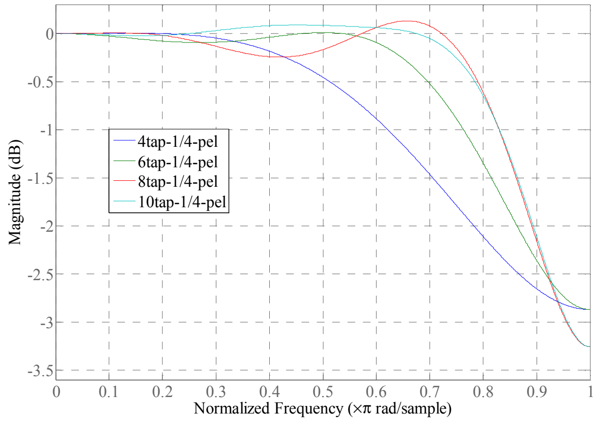
$$i_{0,0} = \left(-4\times a'_{0,-1} + 36\times a'_{0,0} + 36\times a'_{0,1} - 4\times a'_{0,2}\right) \gg shift2 \tag{6b}$$

$$p_{0,0} = \left(-a'_{0,-1} + 15\times a'_{0,0} + 56\times a'_{0,1} - 6\times a'_{0,2}\right) \gg shift2 \tag{6c}$$

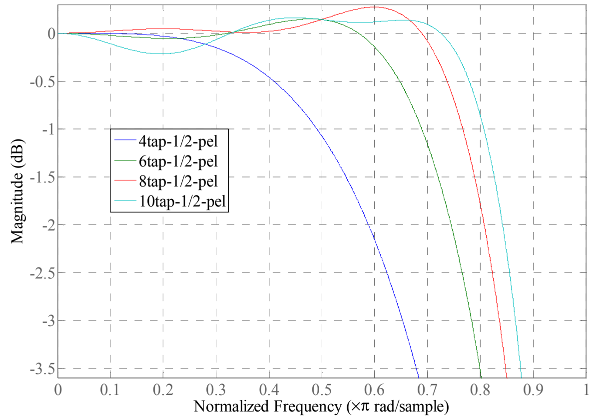
The interpolation processes for different tap filters are similar and 2D separable: for the horizontal fractional positions a, b and c illustrated in [4], horizontal 1D filter is used. For the vertical fractional positions d, h and n, vertical 1D filter is used. For remaining positions, first horizontal 1D filter is applied for extended block and then vertical 1D filter is used. Take 4 tap filter as an example, the fractional-pel pixels $a_{0,0}$, $b_{0,0}$ and $c_{0,0}$ shall be derived by applying the 4 tap filter in horizontal direction to the adjacent integer pixels as described by (5a)~(5c). And the fractional-pel pixels $d_{0,0}$, $h_{0,0}$ and $n_{0,0}$ shall be derived by applying the 4 tap filter in vertical direction, similarly. The fractional-pel pixels $e_{0,0}$, $i_{0,0}$, $p_{0,0}$, $f_{0,0}$, $j_{0,0}$, $q_{0,0}$, $g_{0,0}$, $k_{0,0}$ and $r_{0,0}$ shall be derived by applying the 4 tap filter to the fractional-pel pixels $a_{0,i}$, $b_{0,i}$ and $c_{0,i}$ ($i = -1, ..., 2$) in vertical direction respectively. Equation (6a)~(6c) show the interpolation formulas of pixels $e_{0,0}$, $i_{0,0}$, $p_{0,0}$.

B. *Frequency response analysis*

Fig. 1 illustrates the frequency responses of the filters corresponding to position 1/4, 2/4 and 3/4, and the responses of 1/4 filter and 3/4 filter are the same because of the mirror symmetry of filter coefficients. Note that the frequencies are normalized in the range -1.0 to 1.0, where 1.0 corresponds to half the sampling frequency or pi radians. For the quarter-pel interpolation filters, the passband has smaller ripples and the transition band is much steeper for longer tap filter, but the cut-off frequency turns smaller with the tap increases. So shorter filter will preserve more high frequency details of video sequences for 1/4 and 3/4 filters in theory, and therefore better coding performance can achieved using shorter filter in high-definition video because there are more textures and details. However, for half-pel interpolation filters, the 3dB cut-off frequency becomes bigger with the tap increases. Using longer filter for half-pel interpolation while using shorter filter for quarter-pel interpolations may be a good combination.



(a) 1/4-pel and 3/4-pel interpolation filters



(b) 1/2-pel interpolation filter

Fig. 1. Magnitude response curves of interpolation filters.

C. *Complexity analysis*

Computation complexity is another important aspect for fractional-pel interpolation. We describe the complexity as memory accesses and arithmetic operations, and summarize the complexity results of interpolating each fractional-pel pixel corresponding to different methods. The analysis is necessary and beneficial to SIMD and hardware optimizations.

TABLE V. MAXIMUM PIXEL ACCESSES ANALYSIS

Max	4tap (s=6)	6tap (s=6)	8tap (s=6)	10tap (s=6)
	(L+3)×(W+3)	(L+5)×(W+5)	(L+7)×(W+7)	(L+9)×(W+9)
W=L=4	49	81	121	169
W=L=8	121	169	225	289
W=L=16	361	441	529	625
W=L=32	1225	1369	1521	1681
W=L=64	4489	4761	5041	5329

TABLE VI. MULTIPLICATION OPERATIONS ANALYSIS

Pel	Mults(×W×L)			
	4tap (s=6)	6tap (s=6)	8tap (s=6)	10tap (s=6)
A	0	0	0	0
a	3	5	6	7
b	4	6	6	8
c	3	5	6	7
d	3	5	6	7
e	3×4+3	5×6+5	6×8+6	7×10+7
f	4×4+3	6×6+5	6×8+6	8×10+7
g	3×4+3	5×6+5	6×8+6	7×10+7
h	4	6	6	8
i	3×4+4	5×6+6	6×8+6	7×10+8
j	4×4+4	6×6+6	6×8+6	8×10+8
k	3×4+4	5×6+6	6×8+6	7×10+8
n	3	5	6	7
p	3×4+3	5×6+5	6×8+6	7×10+7
q	4×4+3	6×6+5	6×8+6	8×10+7
r	3×4+3	5×6+5	6×8+6	7×10+7
Avg	10.625	23.0	32.625	48.125

TABLE VII. ADDITION OPERATIONS ANALYSIS

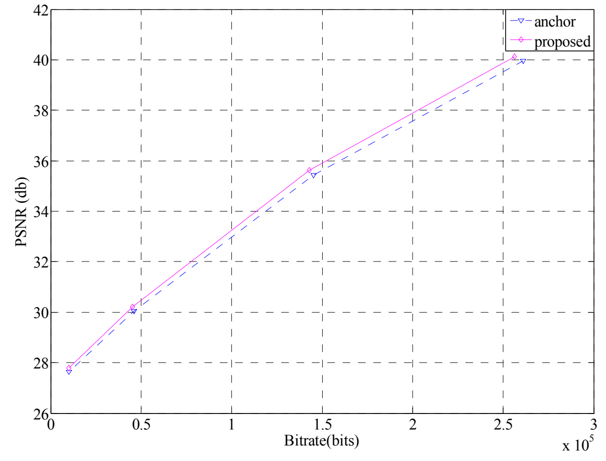
Pel	Adds(×W×L)			
	4tap (s=6)	6tap (s=6)	8tap (s=6)	10tap (s=6)
A	0	0	0	0
a	3	5	7	8
b	3	5	7	9
c	3	5	7	8
d	3	5	7	8
e	3×4+3	5×6+5	7×8+7	8×10+8
f	3×4+3	5×6+5	7×8+7	9×10+8
g	3×4+3	5×6+5	7×8+7	8×10+8
h	3	5	7	9
i	3×4+3	5×6+5	7×8+7	8×10+9
j	3×4+3	5×6+5	7×8+7	9×10+9
k	3×4+3	5×6+5	7×8+7	8×10+9
n	3	5	7	8
p	3×4+3	5×6+5	7×8+7	8×10+8
q	3×4+3	5×6+5	7×8+7	9×10+8
r	3×4+3	5×6+5	7×8+7	8×10+8
Avg	9.5625	21.5626	38.0625	54.69

In fact, a pixel block is regarded as a unit to be processed in terms of different platforms or implementation methods such as SIMD and hardware. Table V~VII give a specific and theoretic calculation to illustrate the complexity of the interpolation including pixel accesses and arithmetical operations. The ratio of maximum pixel accesses of 8 tap interpolation to those of 4 tap interpolation is 2.5 at most and will decrease when the block becomes bigger. For 10 tap interpolation to 8 tap interpolation, the ratio can be more than 1.4. For the arithmetical operations, the average numbers of multiplications and additions of 8 tap are about triple and quadruple as many as those of 4 tap, respectively. For 10 tap to 8 tap the ratio is about 1.5. With the tap decreasing, the computation complexity reduces obviously and considerably.

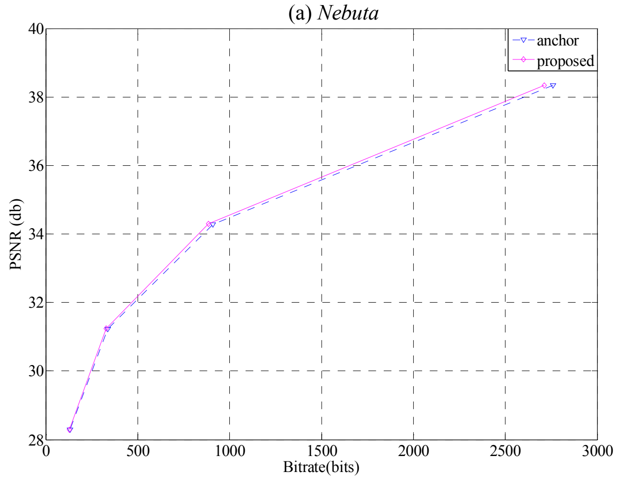
All kinds of the filters are described and analyzed above. Considering both the complexity and coding efficiency, we proposed a resolution-adaptive scheme that the interpolation filter varies along with the resolution of video. More specifically, 4 tap filter is only used for UHD video, and 6 tap, 10 tap are used for the videos larger than and smaller than 720P respectively. The involved coefficients don't need to be coded and consequently no extra bits will increase.

III. EXPERIMENTAL RESULTS

The proposed method is implemented in HEVC reference software HM11.0 [11]. The configuration Low-delay P - Main (LP-Main) [12] is tested, and HM11.0 is used as the anchor for all experiments.



(a) *Nebuta*



(b) *BQSquare*

Fig. 2. Rate-PSNR curves for sequences *Nebuta* and *BQSquare*.

Several experiments corresponding to different tap filters are conducted. According to results in Table VIII, the average BD rate saving increases as the filter tap becomes longer for Class C and D sequences, and it is quite apparent by observing the sequences *PartyScene* and *BQSquare*. However, high-definition videos tend to show the opposite phenomenon, for the BD rate saving increases with the filter tap becomes shorter, such as sequence *Nebuta*. To sum up, the proposed resolution-adaptive scheme obviously has a better coding efficiency than HEVC method. The average BD-rate gains on luma Y, chroma U and V for all the HEVC sequences in Table VIII are 1.4%, 0.7% and 0.7% for LP-Main. Fig. 2 shows the distortion-rate curves for sequences *Nebuta* and *BQSquare*.

IV. CONCLUSION

The generation algorithm of interpolation filter coefficients is introduced in this paper. Different tap filters are compared both in performance and computational complexity, and a resolution-adaptive scheme is proved to be the optimal method for the low complexity and considerable coding efficiency. All the HEVC test sequences with different resolutions are tested, and experiments show that the average BD-rate gains on luma Y, chroma U and V are 1.4%, 0.7% and 0.7%, when the proposed method is conducted in HM11.0. Furthermore, it contributes to making software optimization more effectively especially when SIMD-like operation is used.

ACKNOWLEDGMENT

This work is partially supported by the grants of National Science Foundation of China 61370115, Shenzhen Basic Research Program of JC201104210117A, JC201105170732A, and JCYJ20120614503016, and grants from the Chinese National Natural Science Foundation under contract No.61171139 and No. 61035001, and National High Technology Research and Development Program of China (863 Program) under contract No.2012AA011703.

REFERENCES

[1] ITU-T Recommendation and International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC), Oct. 2004.

[2] Ronggang Wang, Chao Huang, Jintao Li and Yanfei Shen, “Sub-pixel Motion Compensation Interpolation Filter in AVS”, IEEE International Conference on Multimedia and Expo (ICME), Jun. 2004.

[3] Elena Alshina, Alexander Alshin and JeongHoon Park, “CE3: 7 taps interpolation filters for quarter pel position MC from Samsung and Motorola Mobility”, JCTVC-G778, Geneva, CH, Nov. 2011.

[4] Benjamin Bross, Woo-Jin Han, Jens-Rainer Ohm, Gary J. Sullivan, Ye-Kui Wang and Thomas Wiegand, “High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent) ”, JCTVC-L1003, Geneva, CH, Jan. 2013.

[5] T. Wedi, “Adaptive interpolation filter for motion compensated prediction,” in Proc. IEEE Int. Conf. Image Process., vol. 2. Sep. 2002, pp. 509–512.

[6] D. Rusanovskyy, K. Ugur, A. Hallapuro, J. Lainema and M. Gabbouj, “Video coding with low-complexity directional adaptive interpolation filters”, IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 8, pp. 1239-1243, Aug. 2009.

[7] D. Rusanovskyy, K. Ugur and M. Gabbouj, “Adaptive interpolation with flexible filter structures for video coding”, in Proc. IEEE ICIP, Nov. 2009, pp. 1025–1028.

[8] M. Karczewicz, Y. Ye, P. Chen and G. Motta, “Experimental Results of Interpolation Filters on High-Definition Sequences”, ITU-T Q.6/SG16 (VCEG) document VCEG-AJ30, Oct. 2008.

[9] Jie Dong and King Ngi Ngan, “Parametric Interpolation Filter for HD Video Coding”, IEEE transactions on circuits and systems for video technology, vol. 20, no. 12, Dec. 2010.

[10] Youngo Park, Sunyoung Jeon, Chanyul Kim and Kwang Pyo Choi, “Unified design for motion compensation filter”, JCTVC-F480, Torino, IT, Jul. 2011.

[11] Frank Bossen, David Flynn and Karsten Suehring, “JCT-VC AHG report: HEVC HM software development and software technical evaluation (AHG3)”, JCTVC-N0003, Vienna, AT, Jul. 2013.

[12] Frank Bossen, “Common test conditions and software reference configurations”, JCTVC-L1100, Geneva, CH, Jan. 2013.

TABLE VIII. DIFFERENT TAP FILTERS IMPLEMENTED IN HM11.0 FOR LP-MAIN

Resolution	Sequence	4 tap (%)			6 tap (%)			8 tap (%)			10 tap (%)		
		Y	U	V	Y	U	V	Y	U	V	Y	U	V
Class A 2560x1600	Traffic	**0.0**	**-1.4**	**-1.9**	0.5	-0.6	-0.8	2.0	-0.4	-0.6	2.6	0.2	-0.3
	PeopleOnStreet	**-3.0**	**-1.7**	**-1.3**	-0.8	-0.9	-0.6	0.0	-0.4	-0.4	0.8	-0.5	0.1
	Nebuta	**-6.1**	**1.0**	**-0.7**	-1.6	0.7	0.2	-0.7	-0.9	-0.7	0.6	-0.7	-0.7
	SteamLocomotive	**-4.4**	**0.9**	**-0.5**	-2.3	0.0	-0.8	-0.8	0.8	-2.6	-0.7	-0.4	-1.5
Class B 1920x1080	Kimono	-1.5	-0.1	-0.3	**-0.8**	**-0.6**	**-0.7**	-0.2	-0.4	-0.5	0.1	-1.2	-0.7
	ParkScene	2.8	-0.3	-0.7	**0.3**	**-0.9**	**-1.1**	0.8	-0.2	-0.3	1.1	-0.6	-0.3
	BasketballDrive	0.0	-0.6	-0.5	**-0.4**	**-0.5**	**-0.5**	0.0	-0.2	-0.4	0.0	-0.4	-0.2
	BQTerrace	1.9	-1.8	-1.3	**0.3**	**-1.5**	**-0.8**	1.9	-1.1	0.3	2.9	0.2	0.2
Class C 832x480	BasketballDrill	2.2	-2.8	-1.8	1.2	-0.9	-0.4	0.9	-0.3	0.0	**0.1**	**0.3**	**0.8**
	BQMall	4.8	2.4	3.1	1.3	0.1	0.6	-0.1	-0.9	0.4	**0.0**	**-0.8**	**-0.3**
	PartyScene	18.0	9.9	10.0	4.7	1.6	1.9	0.1	-0.7	-0.8	**-1.7**	**-1.1**	**-1.1**
Class D 416x240	BasketballPass	0.7	0.0	0.1	0.0	-0.5	-0.4	-0.2	-0.7	-0.6	**-0.1**	**-0.3**	**0.0**
	BQSquare	36.2	23.0	22.0	10.2	4.9	4.3	0.3	0.3	-0.2	**-2.9**	**-2.2**	**-2.6**
	BlowingBubbles	14.4	6.9	6.9	3.8	0.7	1.1	0.1	-1.0	-0.7	**-1.5**	**-0.7**	**-0.2**
Class E 1280x720	Crew	-1.7	0.3	-0.4	**-1.1**	**-0.4**	**-0.2**	0.4	0.0	0.1	0.6	0.3	0.0
	City	1.5	0.2	0.6	**-0.5**	**-1.4**	**-0.2**	3.3	-0.3	-1.0	4.6	0.1	0.3
	Vidyo3	-4.5	-0.8	-0.2	**-1.5**	**0.1**	**-0.5**	2.1	0.2	0.8	3.3	-0.3	1.4
Class A		**-3.4**	**-0.3**	**-1.1**	-1.0	-0.2	-0.5	0.1	-0.2	-1.1	0.8	-0.3	-0.6
Class B		1.0	-0.4	-0.4	**0.0**	**-0.7**	**-0.6**	0.6	-0.4	-0.2	1.0	-0.4	-0.2
Class C		6.3	2.3	2.9	1.7	0.1	0.4	0.2	-0.5	-0.1	**-0.5**	**-0.5**	**-0.2**
Class D		13.6	7.6	7.5	3.6	1.1	1.4	0.2	-0.4	-0.4	**-1.5**	**-1.0**	**-0.9**
Class E		-1.1	0.3	0.4	**-1.0**	**-0.6**	**-0.3**	1.7	0.0	0.1	2.6	0.4	0.7
Proposed Method		-1.4%(Y), -0.7%(U) and -0.7%(V)											