Block-based Interpolation Filter in Video Coding

MyungJun Kim
Department of Computer Engineering
Sejong University, Digital Media System Laboratory
Seoul, Republic of Korea
mjkim@sju.ac.kr

Yung-Lyul Lee
Department of Computer Engineering
Sejong University, Digital Media System Laboratory
Seoul, Republic of Korea
yllee@sejong.ac.kr

***Abstract*— In high-efficiency video coding (HEVC), fractional pixel motion compensation uses an 8-point filter and a 7-point filter, based on the discrete cosine transform (DCT), for the 1/2-pixel and 1/4-pixel interpolations, respectively. In this paper, discrete sine transform (DST)-based interpolation filters (DST-IFs) and DCT-based interpolation filters (DCT-IFs) are proposed for fractional pixel motion compensation. The DCT-IFs and DST-IFs are applied to the block, coding tree unit (CTU), to decide between DCT-IFs and DST-IFs in terms of rate-distortion optimization (RDO) process. The block-based interpolation filter method showed average Bjøntegaard Delta (BD)-rate reductions of 0.8% and 0.7% in the random access (RA) and low delay B (LDB) configurations, respectively, in HEVC, whereas BD-rate is increased 0.6% in the low delay P (LDP) configuration.**

Keywords*— *HEVC; DCT (discrete cosine transform); DST (discrete sine transform)

I. INTRODUCTION

The International Telecommunication Union Telecommunication Standardization Sector-Video Coding Expert Group (ITU-T VCEG) and the Moving Picture Expert Group (ISO/IEC MPEG) organized the Joint Collaborative Team on Video Coding (JCT-VC)[1], and they jointly developed the next-generation video-coding standard HEVC/H.265. In high-efficiency video coding (HEVC)[2], the motion-compensated prediction (MCP) is a significant video-coding function. It reduces the amount of information which should be transmitted to a decoder by using temporal redundancy in video signals[3–6]. In the MCP, each prediction unit (PU, block) in the encoder finds the best matching block that has the least SAD (sum of absolute difference) from the reference pictures in terms of the Lagrangian cost[7]. Using the block, the motion vector that represents the movement from the current block to the block is transmitted to the decoder with the residual signals that are the difference signals between the current block and the best matching block. Since the moving objects between two pictures are continuous, it is difficult to identify the actual motion vector in block-based motion estimation. Hence, by utilizing fractional accuracy for motion vectors instead of integer accuracy, the residual error is decreased and coding efficiency of video compression is increased. Therefore, the use of fractional pixels that have been derived from an interpolation filter for motion-vector searches can improve the precision of the MCP.

In this paper, we proposed DST-IFs for fractional pixel motion compensation in terms of coding efficiency improvement. DCT-IFs[8] and DST-IF[9]s are applied to the block, coding tree unit (CTU), to decide between DCT-IFs and DST-IFs in terms of rate-distortion optimization (RDO) process. The block-based interpolation filter method showed average BD-rate reductions of 0.8% and 0.7% in the RA and LDB configurations, respectively, in HEVC, whereas BD-rate increment of 0.6% in the low delay P (LDP) configuration.

In this paper, block-based interpolation filter is presented in HEVC. The paper is organized as follows. Section II presents the detail of the proposed method, Section III presents experimental results and Section VI concludes the paper.

II. PROPOSED METHODS

A. Discrete-sine Transform-based Interpolation Filters

The proposed DST-IF for HEVC can easily be designed from the forward/inverse DST-VII. The DST-VII and inverse DST-VII are defined as follows:

$$X(k) = \sqrt{\frac{2}{N+\frac{1}{2}}}\sum_{n=0}^{N-1} x(n)\sin\frac{(n+1)(k+\frac{1}{2})\pi}{N+\frac{1}{2}} \quad (1)$$

$$x(n) = \sqrt{\frac{2}{N+\frac{1}{2}}}\sum_{k=0}^{N-1} X(k)\sin\frac{(n+1/2)(k+1)\pi}{N+\frac{1}{2}} \quad (2)$$

where X(k) is the DST-VII coefficient and x(n) represents the input pixels. The substitution of (1) into (2) results in the following DST-IF equation:

$$x(n) = \frac{2}{N+\frac{1}{2}}\sum_{m=0}^{N-1} x(m)\sum_{k=0}^{N-1}\sin\frac{(m+1)(k+\frac{1}{2})\pi}{N+\frac{1}{2}}\sin\frac{(n+\frac{1}{2})(k+1)\pi}{N+\frac{1}{2}} \quad (3)$$

The DST-IF is derived from (3). For example, the 1/2-pixel interpolation filter, when n = 3.5, in the 8-point DST (N = 8) is derived as a linear combination of the sine coefficients and x(m), m = 0, 1, …, 7. Similarly, the 1/4-pixel interpolation filter, when n = 3.25, in the 7-point DST (N = 7) is derived as a linear combination of the sine coefficients and x(m), m = 0, 1, …, 6. Finally, the DST-IFs with the 1/2-pixel and 1/4-pixel interpolations are shown in Table 2. The filter-coefficient order of the 3/4-pixel interpolation filter is the reverse of the filter-coefficient order of the 1/4-pixel interpolation filter.

978-1-5386-2615-3/18/$31.00 ©2018 IEEE

								7
half-pel filter[i]	-2	6	-13	41	41	-13	6	-2
1/4-pel filter[i]	-2	5	-11	58	18	-6	2	

Figure 1 is an example of the integer- and fractional-pixel positions in the Luma motion compensation. In Figure 1, the capital letters (A_0 to A_7) indicate the integer-pixel position, the small letter b_0 is the 1/2-pixel position, and a_0 and c_0 are the 1/4-pixel and 3/4-pixel positions, respectively. For example, using the DCT-IF, the b_0 and a_0 are calculated from Table 1 as follows:

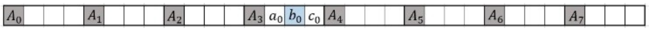

Fig. 1. Fractional position in luma motion compensation

$$b_0 = ((-2)\cdot A_0 + 6\cdot A_1 + (-13)\cdot A_2 + 41\cdot A_3 + 41\cdot A_4 + (-13)\cdot A_5 + 6\cdot A_6 + (-2)\cdot A_7 + 32) \gg \quad (4)$$

where the computation of a_0 is the same as that of b_0 from Table 1, the computation of c_0 is in the order that is the reverse of that of a_0. The “>>” operation means the bit-wise shift right.

B. Block-based Interpolation Filter

DCT-IFs and DST-IFs are applied to the block, coding tree unit (CTU). The proposed method is to compare between the blocks applied DCT-IFs and DST-IFs in terms of BD-rate and determine the least costed block. The decoder receives a flag that which interpolation filter is applied to the block.

III. Experimental Results

The proposed DST-IF is implemented on the HM 16.6. The experiments are performed according to the common test conditions of the HM16.6 (HEVC reference SW). The proposed method is applied to all frames when the QP values are 22, 27, 32, and 37. The classes B, C, and D sequences are tested. The class B, C, and D sequences have resolutions of 1080p, 832x480, and 416x240, respectively. Also, Table II shows the experimental results of bitrate reduction for luma in LDB, LDP, and RA, respectively. The number of frames tested in each sequence is whole frames.

Table II shows the performance comparison of the proposed method to HM 16.6 in terms of BD-rates. Although the block-based interpolation filter has a burden of the flag per CTU, the proposed method shows the average BD-rate reductions of 0.8% and 0.7% in the RA and LDB configurations. Especially for BQSquare which is one of the sequence in Class C has a coding gains result up to 4.7% in the RA configuration. Not only from RA also LDB configuration, the proposed method brings 3.8% of coding gain in BQSquare. However, the average BD-rate increment of 0.6% in LDP configuration. The uni-directional prediction occurs negative gains from applying DST-IFs. As a result, the coding gains of the proposed method is significantly from bi-directional prediction[8]. The proposed method, block-based interpolation filter, shows an increment of 235% of computational complexity in RA configuration. On the other hand, we experimented combining DCT-IFs and DST-IFs methods in CU-level RDO process. The result of this method increased bitrate because of the burden of the flag and also doubled encoding time.

TABLE II. Block-based Interpolation Filter

Classes	Sequences	LDB (%)	LDP(%)	RA(%)
B	Kimono	0.4	0.9	0.6
	ParkScene	0.1	0.7	0.2
C	BasketballDrill	-0.6	0.4	-0.1
	BQMall	-0.5	0.6	-0.4
D	BasketballPass	-0.2	0.3	-0.3
	BQSquare	-3.4	0.4	-4.7
Averages	Class B	0.3	0.8	0.4
	Class C	-0.6	0.5	-0.3
	Class D	-1.8	0.4	-2.6
	All	-0.7	0.6	-0.8

IV. Conclusions

In this paper, we proposed the block-based interpolation filter to yield bitrate reduction. The proposed method decides whether to use the DST-IFs or DCT-IFs in CTU level. The experimental results show that the proposed method achieves a bit rate reduction of 0.8% and 0.7% in RA and LDB configurations, respectively. The other method, BD-rate increment of 0.6% in the LDP.

Acknowledgment

This research was in part supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(Ministry of Science, ICT and Future Planning) (NRF-2015R1A2A2A01006085).

References

[1] Bros, B.; Han,W.-J.; Ohm, J.-R.; Sullivan, G.J.;Wang, Y.-K.;Wiegand, T. High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent). J. Inst. Telev. Eng. Jpn. 2013, 67, 244–247.

[2] Sullivan, G.J.; Ohm, J.-R.; Han,W.-J.;Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) Standard. IEEE Trans. Circuits Syst. 2012, 22, 1649–1668.

[3] Jain, J.R.; Jain, A.K. Displacement measurement and its application in interframe image coding. IEEE Trans. Commun. 1981, 29, 1799–1804.

[4] Girod, B. Motion-compensating prediction with fractional-pel accuracy. IEEE Trans. Commun. 1993, 41,604–612.

[5] Wiegand, T.; Zhang, X.; Girod, B. Long-term memory motion-compensated prediction. IEEE. Trans. Circuits Syst. Video Technol. 1999, 9, 70–84.

[6] Flierl, M.;Wiegand, T.; Girod, B. Rate-constrained multihypothesis prediction for motion-compensated video compression. IEEE Trans. Circuits Syst. Video Technol. 2002, 12, 957–969.

[7] Rosewarne, C.; Bross, B.; Naccari, M.; Sharman, K.; Sullivan, G.J. High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description Update 2; JCTVC-T1002; ITU-T/ISO/IEC Jt. Collab. Team Video Coding (JCT-VC): New York, NY, USA, February 2015.

[8] Kim, M.J.; Kim, N.; Lee, Y. Investigation on interpolation filters in HEVC. In Proceedings of the International Workshop on Advanced Image Technology, Penang, Malaysia, 8–10 January 2017.

[9] Stitch (2013), “Discrete Sine Transform,” http://planetmath.org/sites/default/files/texpdf/39764.pdf (accessed 1 September 2016).