논문 2013-50-10-22

Optimized Implementation of Interpolation Filters for HEVC Encoder

Taejin Hwang, Yongjo Ahn, Jiwoo Ryu, and Donggyu Sim©*

Abstract

In this paper, a fast algorithm of discrete cosine transform-based interpolation filter (DCT-IF) for HEVC (high efficiency video coding) encoder is proposed. DCT-IF filter accounts for around 30% of encoder complexity, according to the computational complexity analysis with the HEVC reference software. In this work, the proposed DCT-IF is optimized by applying frame-level interpolation, SIMD optimization, and task-level parallelization via OpenMP on a developed C-based HEVC encoder. Performance analysis is conducted by measuring speed-up factor of the proposed optimization technique on the developed encoder. The results show that speed-up factors by frame-level interpolation, SIMD, and OpenMP are approximately 38-46, 3.6-4.4, and 3.0-3.7, respectively. In the end, we achieved the speed-up factor of 498.4 with the proposed fast algorithm.

Keywords: HEVC, DCT-IF, Interpolation filter

Ⅰ. Introduction

High efficiency video coding (HEVC) is the latest video coding standard that is characterized by its high compression performance with hierarchical coding structure and other coding tools. Predecessors of HEVC used a basic coding unit of fixed size, while HEVC has the coding tree unit (CTU) of variable sizes[1]. In addition, HEVC employs a variety of advanced coding tools such as 33-directional intra prediction[2], advanced motion vector prediction (AMVP)[3], motion vector merge[4], and sample adaptive offset (SAO)[5] to enhance coding efficiency.

The use of flexible coding unit size and advanced coding tools gives HEVC a great improvement of compression performance (about 50% higher than H.264/AVC). However, it also has high computational complexity, especially for HEVC encoder; such high computational complexity is a major obstruction for commercialization of real-time HEVC encoder.

To reduce computational complexity of HEVC encoder, this paper proposes frame-level interpolation and parallel optimization for HEVC discrete cosine transform-based interpolation filter (DCT-IF), which accounts for 30% of encoder computational complexity in the HM reference software[6]. Frame-level interpolation is proposed to reduce major source of computational load for HEVC encoder and single instruction multiple data (SIMD) implementation and parallelization via OpenMP are proposed for further acceleration.

This paper is organized as follows. Section Ⅱ introduces HEVC DCT-IF and Section Ⅲ presents the proposed optimization techniques. In Section Ⅳ,

* Dept. of Computer Engineering, Kwangwoon University
© Corresponding Author(E-mail: dgsim@kw.ac.kr)
※ This work was supported by the grant from the Seoul R&BD Program (SS110004M0229111)
접수일자: 2013년9월9일, 수정완료일: 2013년9월26일

the proposed techniques are evaluated with several experiments. In Section Ⅴ, a conclusion and further research topic are given.

Ⅱ. DCT-IF of HEVC

The HEVC inter-frame prediction uses fractional-pel motion estimation (ME) for high prediction performance. In fractional-pel ME, motion vectors can be represented in 1/4 pixel accuracy for luma and 1/8 pixel accuracy for chroma, which require interpolation operations for luma and chroma components according to their accuracy. DCT-IF of HEVC is designed with 8-tap (7-tap) filter for luma and 4-tap filter for chroma. The filter coefficients are shown in Table 1.

Table 1. Interpolation filter coefficients of HEVC.

Position		Filter coefficient
Luma	1/4	-1, 4, -10, 58, 17, -5, 1, 0
	2/4	-1, 4, -11, 40, 40, -11, 4, -1
	3/4	0, 1, -5, 17, 58, -10, 4, -1
Chroma	1/8	-2, 58, 10, -2
	2/8	-4, 54, 16, -2
	3/8	-6, 46, 28, -4
	4/8	-4, 36, 36, -4
	5/8	-4, 28, 46, -6
	6/8	-2, 16, 54, -4
	7/8	-2, 10, 58, -2

Based on profiling of the HEVC reference software, inter-frame prediction takes up 80% of encoder complexity, and the computational complexity of DCT-IF accounts for 37.5% of inter-frame prediction (30% of the total encoder complexity)[6]. High computational complexity of DCT-IF is due to the numbers of filter coefficients (Interpolation filter of H.264/AVC has 6-tap at most), which lead to more memory access and arithmetic operations. DCT-IF is considered as a one of the most computationally intensive processes in HEVC encoders[6].

Ⅲ. The proposed optimization techniques for DCT-IF in HEVC encoder

This paper proposed optimization techniques for DCT-IF. The proposed techniques consist of frame-level interpolation, SIMD optimization, and data-level parallelization with OpenMP.

3.1 Frame-level interpolation filter

Motion estimation in HEVC finds the most correlating block within a search range in the reference picture for the current PU. The same fractional-pel in the reference picture should be referenced multiple times for different PUs, as the search ranges of the PUs are largely overlapped each other. HM performs redundant interpolations for the same fractional-pel position in such cases. Moreover, fractional-pels computed in ME are not used for motion compensation (MC). Thus, redundant DCT-IF is needed for the MC.

To eliminate superfluous interpolations in ME and MC, the proposed frame-level interpolation is performed on the reference frame. While HM performs interpolation on the fly for each PU, frame-level interpolation compute all the fractional pels of an entire frame prior to ME and MC. As a result, redundant DCT-IF can be drastically reduced.

3.2 SIMD implementation of DCT-IF

Hardware architectures supporting a SIMD instruction set such as MMX, SSE and SSE2 can process multiple data in a single instruction. SSE and SSE2 instruction sets have arithmetic operations with 128-bit register operands each of which contains eight 16-bit data or four 32-bit data. For efficient code level optimization with SIMD, use of registers withconsiderationfordynamicrangeofinput/outputdataisr equired.

The process of DCT-IF involves a great deal of repeated arithmetic operations, which makes it suitable for SIMD-based code level optimization.

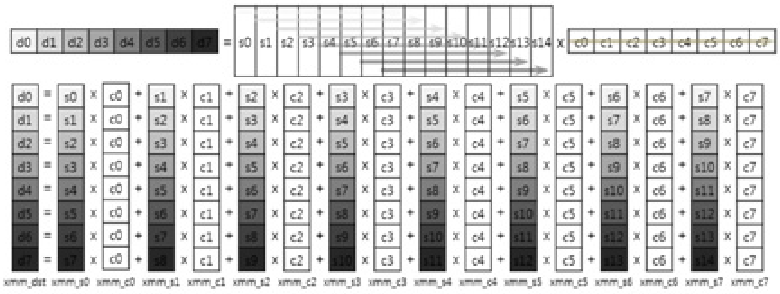

Fig. 1. SIMD implementation of 1-D horizontal filter.

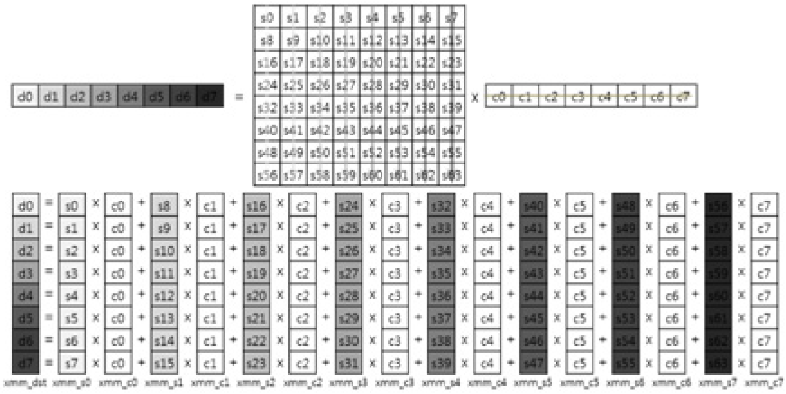

Fig. 2. SIMD implementation of 1-D vertical filter.

DCT-IF in HEVC applies sequential 1-D filterings on horizontal and vertical directions instead of 2-D filtering, for sake of computational complexity. Both inputs and outputs of DCT-IF represent intensity level of pixels, which typically has 8-bit accuracy.

However, intermediate values during the DCT-IF process have increased bit accuracy. For instance, in the series of two 1-D interpolations, the 16-bit output of the first horizontal 1-D filter is directly used as an input of the second vertical 1-D filter without shifting it down to 8 bits. Therefore, the proposed SIMD implementation performs filtering for eight pels in parallel with 128-bit registers.

Figure 1 depicts the proposed SIMD implementation for of 1-D horizontal filter. Arrays denoted s0~s14, d0~d7 and c0~c7 represents inputs, outputs and coefficients of interpolation filtering for eight pixels from the same horizontal line. 15 input pixels (s0~s14) are loaded with a single memory access, and distributed to eight registers accordingly (s0~s7, s1~s8 and so on). Then, the eight output pixels (d0~d7) are concurrently computed with the eight registers and the coefficients.

Vertical filtering works in somewhat different manner, as shown in Fig. 2. Instead of filtering each vertical line separately, the proposed implementation processes eight pels from each vertical lines at the same horizontal position to avoid vertical memory access.

3.3 DCT-IF parallelization using OpenMP

OpenMP is a widely-used technique for multithread parallelization. DCT-IF can be implemented in high parallelism, because there are no

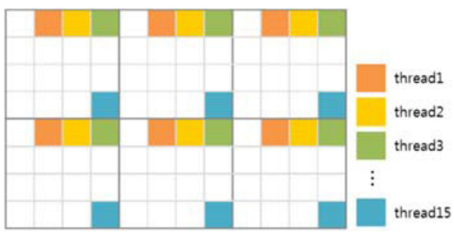

Fig. 3. Parallel DCT-IF.

data dependency among 15 fractional pixels for each integer pixel. In this paper, 15 threads are used to generate fractional pixels in each of 15 fractional-pel positions in a frame, as shown in Fig. 3. For comparative study, parallelization via OpenMP is applied to the proposed frame-level interpolation algorithm with SIMD and without SIMD.

Ⅳ. Experiment results

Eight test sequences are used from Class B and Class C of common test condition (CTC) sequences for standardization. The test sequence set is shown in Table 2. The proposed techniques are implemented on developed HEVC encoder in ANSI C. All SIMD instructions are implemented with intrinsic functions. To evaluate individual effect of each optimization techniques (frame-level interpolation, SIMD and OpenMP), evaluation was performed with different subsets of the optimization techniques.

Table 3 shows speed-up factors of the proposed implementation. The experiment was performed with different sets of optimization techniques to analyze performance of each optimization technique and how they interact each other. For performance evaluation, speed-up factors are used as objective measurements. DCT-IF speed-up factor measures how much times DCT-IF of the proposed implementation faster than an anchor with no optimization. The first column of

Table 2. Test sequence set.

Sequence		Number of frames	Frame rate(fps)	Encoded frames
B Class (1920 ×1080)	BasketballDrive	128	50	144
	BQTerrace	128	60	192
	Cactus	128	50	144
	ParkScene	128	24	72
C Class (832× 768)	BasketballDrill	128	50	144
	BQMall	128	60	192
	RaceHorsesC	128	30	96
	PartyScene	128	50	144

Table 3. Performance of proposed implementation.

Algorithm	IF Time (ms)	IF Speed up
Original IF	1672	-
SIMD IF	475	3.66
Frame-level IF	42	38.06
Frame-level SIMD IF	10	166.68
Parallel Frame-level IF	12	139.89
Parallel Frame-level SIMD IF	3	498.38

Table 3 indicates whether the frame-level interpolation and/or OpenMP are used. “IF”, “Frame-level” and “Parallel frame-level” denotes neither, only frame-level interpolation, and both of two optimization techniques are used, respectively. The second column in Table 3 shows whether SIMD optimization is used or not.

As shown in Table 3, the speed-up factors of the proposed DCT-IF with frame-level interpolation, SIMD, and OpenMP are approximately 38∼46, 3.6∼4.4, and 3.0∼3.7, respectively. Note that the speed-up of SIMD implementation on frame-level IF and frame-level IF is 3.66 and 4.38, respectively. It is observed that frame-level interpolation and parallelization techniques (SIMD and OpenMP implementations) have synergy effect on the overall performance. Applying all proposed optimization techniques (frame-level interpolation, SIMD and OpenMP) speed up DCT-IF by a factor of 498.38.

Ⅴ. Conclusion

This paper proposes an optimized implementation for DCT-IF in HEVC which takes up 30% of HM encoder complexity. The developed algorithm consists of frame-level interpolation and parallel optimization with SIMD and OpenMP. Performance evaluation of the proposed DCT-IF on the developed HEVC encoder shows that speed-up factors of the frame-level interpolation, SIMD and OpenMP are as high as approximately 38∼46, 3.6∼4.4, and 3.0∼3.7, respectively. Total speed-up factor with all the proposed techniques is achieved by 498.38.

Further study for DCT-IF optimization aims to implement a real-time HEVC encoder with SIMD optimization with assembly level optimization as well as shared memory parallelization via CUDA.

REFERENCES

[1] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
[2] F. Bossen, T. K. Tan, and J. Takiue, "Simplified angular intra prediction," document JCTVC-B093, Geneva, CH, July 2010.
[3] I.-K. Kim and T. Lee, "Improved motion vector predictor selection in AMVP," document JCTVC-D337, Daegu, Korea, Jan. 2011.
[4] H. Nakamura, S. Fukushima, and M. Nishitani, "Unification of derivation process for merge mode and MVP," document JCTVC-F419, Torino, IT, July 2011.
[5] C. Fu, C. Chen, C. Tsai, Y. Huang, and S. Lei, "CE13: Sample Adaptive Offset with LCU-Independent Decoding," document JCTVC-E049, Geneva, CH, March 2011.
[6] Y.-J. Ahn, T.-J. Hwang, S.-E. Yoo, W.-J. Han, and D.-G. Sim, "Statistical characteristics and complexity analysis of HEVC encoder software," Journal of Broadcasting & Electronic Media, vol. 17, no. 6, pp. 1091-1105, Nov. 2012.