Performance Analysis of HEVC Parallelization Methods for High-Resolution Videos

Hochan Ryu, Yong-Jo Ahn, Jung-Soo Mok, and Donggyu Sim

Department of Computer Engineering, Kwangwoon University / Seoul, South Korea
{jolie, yongjoahn, jsmok, dgsim}@kw.ac.kr

***** Corresponding Author: Donggyu Sim

Received April 5, 2014; Revised June 13, 2014; Accepted November 19, 2014; Published February 28, 2015

** Regular Paper*

***Abstract*:** Several parallelization methods that can be applied to High Efficiency Video Coding (HEVC) decoders are evaluated. The market requirements of high-resolution videos, such as Full HD and UHD, have been increasing. To satisfy the market requirements, several parallelization methods for HEVC decoders have been studied. Understanding these parallelization methods and objective comparisons of these methods are crucial to the real-time decoding of high-resolution videos. This paper introduces the parallelization methods that can be used in HEVC decoders and evaluates the parallelization methods comparatively. The experimental results show that the average speed-up factors of tile-level parallelism, wavefront parallel processing (WPP), frame-level parallelism, and 2D-wavefront parallelism are observed up to 4.59, 4.00, 2.20, and 3.16, respectively.

Keywords: HEVC decoder, Parallelism, Tile, WPP, Wavefront

1. Introduction

Aideo services have evolved rapidly with the demands of high-quality videos. Higher resolution and higher bitrates are required with the evolution of video services. In the current market, there are many video streaming services and the market requires high-resolution video, such as Full HD and 4K-UHD. To correspond to the requirements, the Joint Collaborative Team Video Coding (JCT-VC) of the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) developed High Efficiency Video Coding (HEVC), which is the next generation video coding standard [1]. HEVC is based on quad-tree based coding tree units (CTUs) and coding units (CUs), which can be larger than macroblocks of the previous video coding standards. In addition, the prediction unit (PU) and transform unit (TU) that can be split from CU are used. In addition, 8-tap filters are used to interpolate the fractional samples, compared to 6-tap filtering of H.264/AVC [2]. 35 modes are then used in the intra prediction [3] and sample adaptive offset (SAO) is used in the in-loop filtering process [4]. These new features and additional tools compared to the previous video coding standards increase the decoder complexity [5]. To solve these problems, many studies have focused on fast HEVC encoders and decoders [6, 7]. However, it is difficult to make real-time codecs without parallelization. Therefore, parallelization of the video decoder could be essential for the real time decoding of high resolution videos. Currently, the multi-core system can be used for the parallelization of a video decoder. There are several conventional parallelization methods for video decoders [8, 9]. For efficient parallelization, an objective comparison of the parallel methods on the multi-core system is very important.

In this paper, several parallelization methods of HEVC were analyzed and evaluated comparatively. In the process of establishing a HEVC standard, several techniques were proposed for the parallelism. Finally, the tile and wavefront parallel processing (WPP) were adopted for the HEVC standard. The tile is a rectangular region of coding tree blocks (CTBs) within a particular tile column and a particular tile row in a picture [10]. There are no data dependencies among the partitioned regions. Therefore, the partitioned regions can be decoded in parallel [11]. WPP is a parallelization technique that uses a separate CABAC context for each CTU line. Based on this method, entropy decoding of each CTU line can be conducted

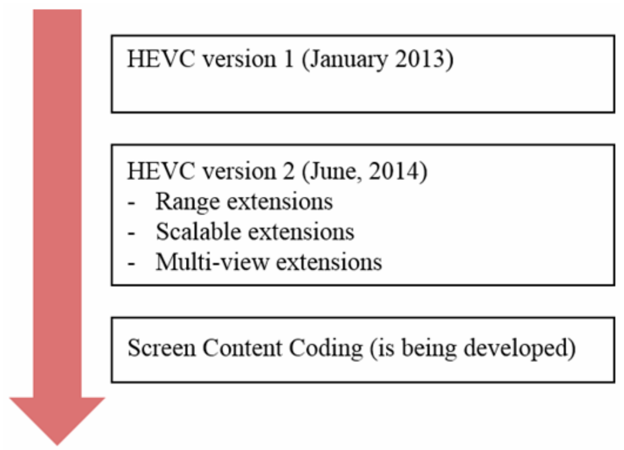


Fig. 1. Standardization flow of HEVC.

simultaneously [12]. Another parallelization method of a HEVC decoder is frame-level parallelization. Using frame-level parallelization, several pictures can be decoded in parallel. On the other hand, for the parallel decoding of each picture, consideration of the structure of reference pictures is necessary. The parallelization methods introduced above are encoder-dependent methods. On the other hand, 2D-wavefront method is not an encoder-dependent method. For this reason, a 2D-wavefront method is used most widely as a parallelization method in video decoders. Using 2D-wavefront parallelization, pixel decoding of the entropy decoded CTUs can be conducted simultaneously in the CTU line level. This paper compares several parallelization methods and evaluates parallel performance of all the parallelization methods objectively.

This paper is organized as follows. Section 2 reviews the HEVC standard. In Section 3, the parallelization methods of HEVC decoder are introduced. The advantages and drawbacks of the methods are then compared. In Section 4, all the parallelization methods are evaluated in parallel and RD performance is assessed. Finally, Section 5 concludes the paper.

2. HEVC Overview

HEVC is the newest video coding standard jointly developed by the ISO/IEC MPEG and ITU-T VCEG focused on significantly better coding efficiency. The first version of the HEVC standard was completed on January, 2013. The second version of the HEVC standard was finalized on June, 2014 including range extensions, scalable coding extensions, and multi-view extensions [14]. In addition, the screen content coding that is designed for computer generated images and videos is being developed [15].

HEVC has new features for the efficiency of high-resolution and high-quality videos. First, HEVC employs hierarchical variable blocks of CU, PU, and TU; whereas, the previous video codecs use the fixed size macroblock [16]. Because the blocks of HEVC can be larger than the previous macroblock, HEVC can compress the high-resolution videos more efficiently. Fig. 2 presents one of

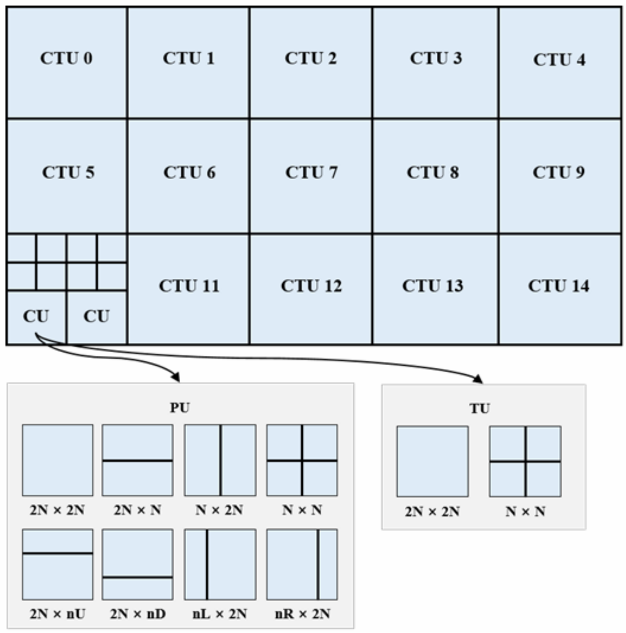


Fig. 2. Block partition of HEVC.

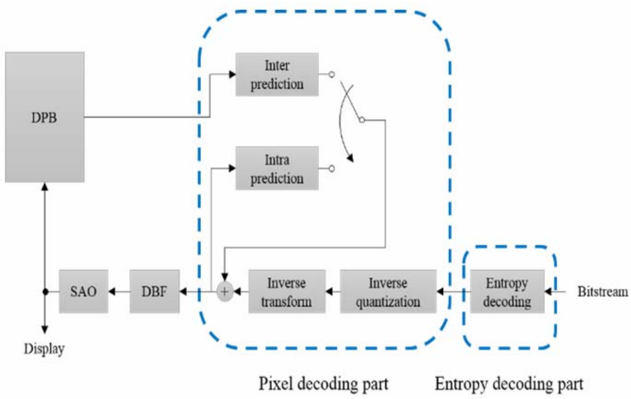


Fig. 3. Block diagram of the HEVC decoder.

the block partition examples based on HEVC. A picture can be partitioned with multiple CTUs and each CTU splits into multiple CUs of that size is variable. In addition, PU and TU are split from each CU.

Secondly, a more accurate intra prediction can be performed with 35 intra prediction modes. In the case of inter prediction, the precision of motion compensation is increased using the 8-tap interpolation filter. Merge and advanced motion vector prediction (AMVP) are used for efficient syntax in representing the inter prediction. Third, HEVC uses an additional in-loop filter called SAO to improve the subjective quality and coding efficiency at the same time. On the other hand, these additional features make the encoder and decoder more complex [17]. The main sources of the HEVC decoder complexity are entropy decoding and pixel decoding parts, as shown in Fig. 3. For high-resolution and high-bitrate videos, the complexity of the entropy decoding part is more complicated [18]. Several parallelization methods can be used to reduce the complexity of entropy decoding and pixel decoding parts.

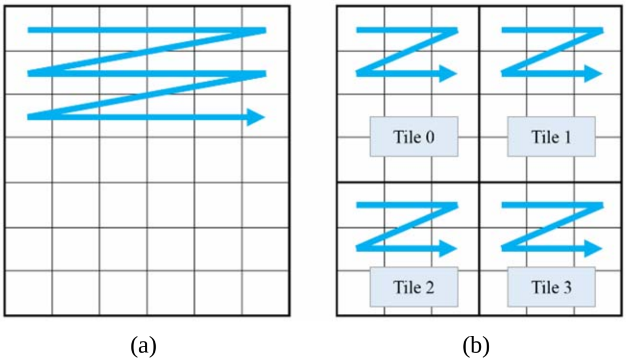


Fig. 4. Example of tile partitioning in a picture (a) No tile splits, (b) Use of 4 tile splits.

Table 1. BD rates according to the number of tiles.

Sequence	BD-rate	
	4 tiles	8 tiles
BasketballDrill	2.03	3.09
BQMall	0.85	1.75
PartyScene	0.60	0.80
Racehorse	0.99	1.34
Avg.	1.12	1.75

3. Parallelization Methods of HEVC Decoder

The HEVC standard is designed to be suitable for parallel processing. Tile and WPP were adopted for better parallel processing. In addition, HEVC supports a parallel friendly in-loop filter structures [13]. In this section, four parallelization methods that can be used in HEVC decoders for the real-time decoding of high-resolution videos are introduced and compared.

3.1 Tile-level Parallelism

Tile-level parallelism can be used for parallel decoding in a picture. A picture can be partitioned into a number of rectangular blocks composed of CTBs.

Fig. 4 gives an example of tile partitioning. No data dependencies exist among the partitioned regions. In addition, the scan order of the CTUs is changed to the tiles scan order. Each tile can be decoded independently by splitting into multiple cores or threads. Therefore, this method is suitable for multi-core-based parallelization. The parallel performance increases with increasing number of tile splits; however, an increase in the tile splits depletes the compression performance. Table 1 lists the compression performance degradation according to the number of tiles. The BD rate loss of 4 tiles and 8 tiles are 1.12% and 1.75% on average.

3.2 Wavefront Parallel Processing (WPP)

WPP is an adopted parallelization method of the HEVC standard. Using this method, the entropy decoding of each CTU line can be conducted in parallel. At the first CTU of

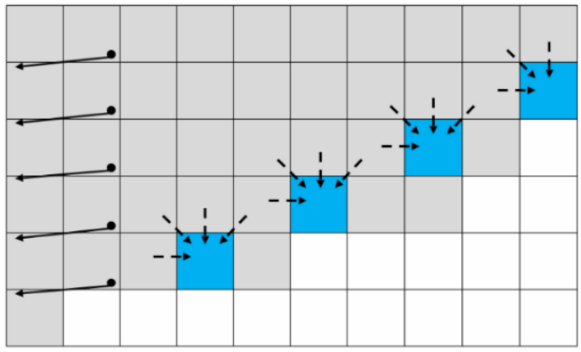


Fig. 5. CABAC context model initialization positions and dependencies of among CTUs.

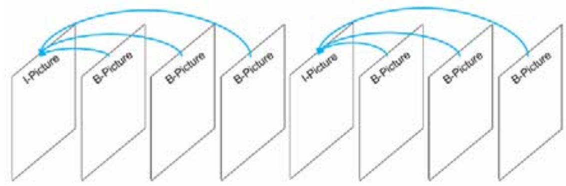


Fig. 6. Example of the GOP structure for the frame-level parallelism.

each CTU line, the CABAC context model is copied from the updated CABAC context that is located at the right-top CTU. If WPP is used, the efficiency of CABAC can be degraded slightly. In addition, the dependence among the CTUs should be considered. Fig. 5 shows the CABAC context model initialization positions and the dependencies of the WPP method. Because the CABAC contexts should be initialized at the first CTU of each CTU line, the entropy decoding of each CTU line cannot begin at the same time. In addition, the dependencies among the CTUs should be checked in a pixel decoding process.

3.3 Frame-level Parallelism

Frame-level parallelism is suitable for the special purpose applications. For the frame-level parallelism, a special GOP structure should be considered in an encoder. Fig. 6 gives an example of the GOP structure for frame-level parallelism. In Fig. 6, all the B-pictures in a GOP only refer to the previous I-picture. Using this structure, the following B-pictures can be decoded in parallel. On the other hand, a limited reference picture can lead to a rapid decrease in compression performance. If the number of B-pictures that refer to previous I-pictures are increased, good parallel performance can be achieved in the decoding parts of the B-pictures. On the other hand, complexity of the I-Picture is much higher than that of the B-picture. Therefore, this parallel method cannot beat the decoding time of the I-picture that is decoded sequentially.

3.4 2D-wavefront Parallelism

2D-wavefront parallelism is the most widely used parallel method in video decoders. This method is an encoder independent method compared to the above three methods. Using 2D-wavefront parallelism, the pixel

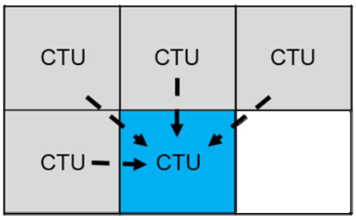


Fig. 7. Dependencies among the neighboring CTUs in pixel decoding.

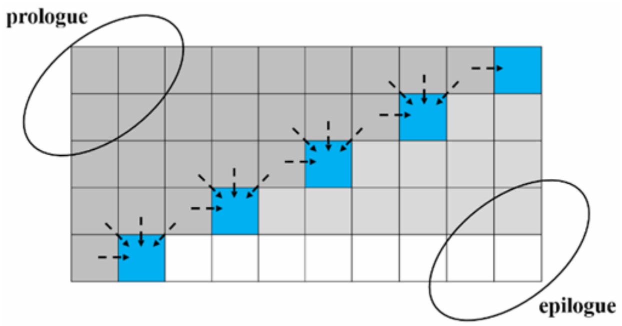


Fig. 8. Problem of 2D-wavefront parallelism.

decoding process can be conducted simultaneously on a CTU line level. Like the WPP method, the dependencies among the CTUs should be considered. Fig. 7 presents the neighboring CTUs that should be decoded before decoding the current CTU.

Fig. 8 shows the problem of 2D-wavefront parallelism. Because of the dependencies of the neighboring CTUs, idle threads can occur in the prologue and epilogue parts in a picture. The maximum number of CTUs (P_CTU) that can conduct pixel decoding in parallel is calculated by Eq. (1). 〚nCTU〛 _h is the number of CTUs in the picture height and 〚nCTU〛 _w is the number of CTUs in the picture width.

$$P_{CTU} = \begin{cases} min(nCTU_h, \dfrac{nCTU_w}{2} + 1, \text{ if } nCTU_w \text{ is odd} \\ min(nCTU_h, \dfrac{nCTU_w}{2} \quad \text{otherwise} \end{cases} \tag{1}$$

In a Full HD resolution, there are 30×17 CTUs that the size is 64×64 in a picture. In this case, the number of CTUs that can be decoded simultaneously are represented in Fig. 9. In the prologue and epilogue parts of a picture, the CTUs that can be decoded at the same time are insufficient. For this reason, the threads can be idle in these parts.

3.5 Comparisons of parallelism

As introduced above, all the parallelization methods have advantages and disadvantages. Table 2 compares the four parallel methods. Note that tile-level parallelism, WPP and frame-level parallelism have compression performance degradation, and these methods are encoder dependent. On the other hand, the 2D-wavefront parallelization method has no degradation of compression performance and it is not dependent on the encoders.

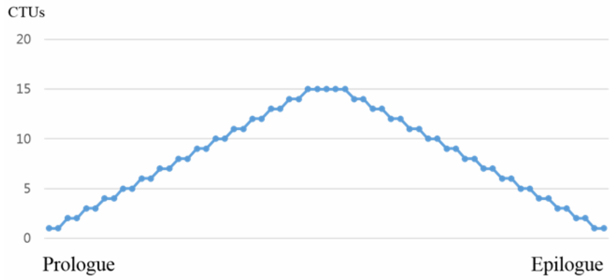


Fig. 9. Distribution of the CTUs that can be decoded at the same time.

Table 2. Comparisons of the four parallel methods.

Parallelization method	Compression performance degradation	Encoder dependency
Tile-level parallelism	○	○
WPP	○	○
Frame-level parallelism	○	○
2D-wavefront parallelism	None	None

Table 3. Experimental conditions.

Processor	Intel Xeon CPU E5-2687W v2 3.40GHz
OS	Windows Server 2008 R2 Standard
Compiler	Intel C++ Compiler XE 13.0
Parallel tool	OpenMP 2.0
Resolution	FHD (1920×1080), 4K-UHD (3840×2160)
Coding mode	Low Delay
QP	22, 27, 32, 37

4. Performance Evaluation

For a performance evaluation of the four parallelism methods, all the parallelization methods were implemented on ANSI C-based private decoder developed based on the HM12.0 decoder [19]. For the parallel implementation, OpenMP was used with Microsoft Visual Studio 2012 [20]. Table 3 lists the experimental conditions.

The parallelism performance was measured by Eq. (2). $Decoding\ time_{ref}$ represents the decoding time of the HM12.0 decoder without any parallelism methods. $Decoding\ time_{prop}$ denotes the decoding time of the parallelized private decoder with a parallelization method.

$$Speed-up = \frac{Decoding\ time_{ref}}{Decoding\ time_{prop}} \tag{2}$$

Table 4 lists the speed-up factors of tile-level

Table 4. Speed-up factors of tile-level parallelism.

Resolution	Sequence	Speed-up according to the number of tiles		
		2	4	6
FHD	BasketballDrive	2.81	3.61	4.45
	BQTerrace	2.37	4.08	4.80
	Cactus	3.04	3.68	4.26
	Kimono1	3.00	4.14	4.65
	ParkScene	3.12	3.84	4.78
	Avg.	2.87	3.87	4.59
4K-UHD	Beauty	2.89	4.15	4.56
	Jockey	3.06	4.02	4.87
	ShakeNDry	2.97	3.64	4.26
	Avg.	2.97	3.94	4.56

Table 5. Speed-up factors of WPP.

Resolution	Sequence	Speed-up according to the number of threads		
		2	4	6
FHD	BasketballDrive	2.47	3.07	3.40
	BQTerrace	2.09	2.93	3.07
	Cactus	2.49	2.95	3.73
	Kimono1	2.10	2.80	2.82
	ParkScene	1.98	2.88	2.85
	Avg.	2.23	2.93	3.17
4K-UHD	Beauty	2.71	3.58	4.60
	Jockey	2.67	3.66	4.11
	ShakeNDry	2.11	2.63	3.28
	Avg.	2.50	3.29	4.00

Table 6. Speed-up factors of frame-level parallelism.

Resolution	Sequence	Speed-up according to GOP size		
		4	6	8
FHD	BasketballDrive	2.12	2.21	2.33
	BQTerrace	2.01	2.13	2.37
	Cactus	1.96	2.06	2.28
	Kimono1	1.71	1.83	1.93
	ParkScene	1.82	1.94	2.07
	Avg.	1.92	2.03	2.20
4K-UHD	Beauty	1.68	1.85	1.89
	Jockey	1.71	1.83	1.85
	ShakeNDry	1.67	1.84	1.91
	Avg.	1.69	1.84	1.88

Table 7. Speed-up factors of 2D-wavefront parallelism.

Resolution	Sequence	Speed-up according to the number of threads		
		2	4	6
FHD	BasketballDrive	2.75	3.43	3.47
	BQTerrace	3.00	3.61	3.67
	Cactus	2.13	2.50	2.47
	Kimono1	2.81	3.50	3.57
	ParkScene	2.17	2.62	2.63
	Avg.	2.57	3.13	3.16
4K-UHD	Beauty	2.04	1.94	2.03
	Jockey	2.36	3.30	2.93
	ShakeNDry	2.03	2.39	2.35
	Avg.	2.14	2.54	2.44

parallelism. The speed-up factors increase with increasing number of tiles. The results shows that the average of the speed-up factors are 4.59 and 4.56 for FHD and 4K-UHD with 6 tiles, respectively.

Table 5 lists the speed-up factors using the WPP. The results show that the speed-up factors for the 4K-UHD sequences are higher than those for FHD sequences. The speed-up factors are influenced by the entropy decoding complexity between the FHD and 4K-UHD sequences. The entropy decoding part of the 4K-UHD sequences is more complex than the entropy decoding of the FHD sequences. Therefore, the parallel performance of the WPP for parallel entropy decoding showed reasonably good parallel performance for the 4K-UHD sequences.

Table 6 presents the results of frame-level parallelism. The speed-up factors are not large. These results are due to the decoding overhead of the I-picture. Table 7 lists the speed-up factors of the 2D-wavefront parallelism. The mean speed-up factors for the FHD videos were higher than those for the 4K-UHD videos. This means that the parallel performance of 2D-wavefront parallelism is affected by the entropy decoding part that is not a parallel region based on the 2D-wavefront parallelism.

5. Conclusion

This paper introduced several parallel methods for HEVC decoders and evaluated the real-time decoding of high-resolution videos. Tile-level parallelism, WPP, frame-level parallelism, and 2D-wavefront parallelism were implemented on the ANSI C-based private decoder. The parallel performance of several parallel methods were compared. For the FHD test sequences, the mean speed-up factors of tile-level parallelism, WPP, frame-level parallelism, and 2D-wavefront parallelism were evaluated and found to be up to 4.59, 3.17, 2.20, and 3.16, respectively. In addition, the speed-up factors of the 4K-UHD sequences are observed up to 4.56, 4.00, 1.88, and 2.54 for tile-level parallelism, WPP, frame-level parallelism, and 2D-wavefront parallelism, respectively.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(NRF-2014R1A2A1A11052210)

References

[1] High Efficiency Video Coding, Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan 2013. Article (CrossRef Link)

[2] J. Lainema, F. Bossen, W. Han, J. Min, and K. Ugur, "Intra Coding of the HEVC Standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp.1792-1801, Dec. 2012. Article (CrossRef Link)

[3] G.J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp.1648-1667, Dec. 2012. Article (CrossRef Link)

[4] C. Fu, E. Alshina, A. Alshin, Y. Huang, C. Chen, and C. Tsai, C. Hsu, S. Lei, J. Park, and W. Han, "Sample adaptive offset in the HEVC standard," *IEEE trans. on Circuits and Systems for Video Technol.*, vol. 22, no. 12, pp. 1755-1764, Dec. 2012. Article (CrossRef Link)

[5] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T.K. Tan, and T. Wiegan, "Comparison of the coding efficiency of video coding standards-Including High Efficiency Video Coding (HEVC)," *IEEE Trans. Circuits Syst. Video technol.*, vol. 22, no. 12, pp. 1668-1683, Dec. 2012. Article (CrossRef Link)

[6] Y. Ahn, T. Hwang, D. Sim, and W. Han, "Implementation of fast HEVC encoder based on SIMD and data-level parallelism," *EURASIP Journal on Image and Video Processing*, March. 2014. Article (CrossRef Link)

[7] E. Ryu, J. Nam, S. Lee, H. Jo, and D. Sim, "Sample adaptive offset parallelism in HEVC," *International Conference on Green and Human Information Technology (ICGHIT 2013)*, Hanoi, Vietnam, Feb. 27 ~ Mar. 1, 2013. Article (CrossRef Link)

[8] H. Jo and D. Sim, "Hybrid Parallelization for HEVC Decoder," *Image and Signal Processing (CISP).*, vol. 1, pp. 170-175, Dec. 2012. Article (CrossRef Link)

[9] E. Baaklini, H. Sbeity, and S. Niar, "H.264 Macroblock Line Level Parallel Video Decoding on Embedded Multicore Processors," *Digital System Design (DSD)*, pp. 902-906, Sep. 2012. Article (CrossRef Link)

[10] B. Bross, W.-J, Han, G.J. Sullivan, J.-R. Ohm, and T. Wiegand, "High Efficiency Video Coding (HEVC) *Text Specification Draft 10 (for FDIS & Last Call,"document JCTVC-L1003_v34*, Geneva, CH, Jan. 2013. Article (CrossRef Link)

[11] K. Misra, A. Segall, M. Horowiz, S. Xu, A. Fuldseth and M. Zhou, "An Overvidw of Tiles in HEVC," *IEEE Selected Topics in Signal Processing*, vol. 7, pp. 969-977, Nov. 2013. Article (CrossRef Link)

[12] C. Chi, M. Alvarez-Mesa, B. Juulink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel Scalability and Efficiency of HEVC Parallelization Approaches," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1827-1838, Dec. 2012. Article (CrossRef Link)

[13] A. Norkin, G. Bjøntegaard, A. Fuldseth, M. Narroschke, M. Ikeda, K. Andersson, M. Zhou, and G. Auwera, "HEVC deblocking filter," IEEE trans. on Circuits and Systems for Video Technol., vol. 22, no. 12, pp. 1746-1754, Dec. 2012. Article (CrossRef Link)

[14] J. Boyce, et al., "Draft high efficiency video coding (HEVC) version 2, combined format range extensions (RExt), scalability (SHVC), and multi-view (MV-HEVC) extensions," *JCTVC-R1013, 18th JCT-VC meeting*, Sapporo, JP, June 2014. Article (CrossRef Link)

[15] ITU-T Q6/16 and ISO/IEC JTC1/SC29/WG11 document N14175, "Joint Call for Proposals for Coding of Screen Content," San Jose**,** USA, Jan. 2014. Article (CrossRef Link)

[16] T. Wiegand, G. J. Sullivan, G.Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003. Article (CrossRef Link)

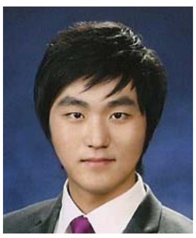
[17] F. Bossen, B. Bross, K. S¨uhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE trans. on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1685-1696, Dec. 2012. Article (CrossRef Link)

[18] H. Jo and D. Sim, "Bitstream decoding processor for fast entropy decoding of variable length coding–based multiformat videos," *Optical Engineering on Imaging Components, Systems, and Processing*, vol. 53, no. 6, June 2014. Article (CrossRef Link)

[19] HM-12.0, https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-12.0 Article (CrossRef Link)

[20] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *IEEE Computational Science & Engineering*, vol. 5, no. 1, pp.46-55, Jan. 1998. Article (CrossRef Link)

Hochan Ryu received his B.S. degree in Computer Engineering from Kwang-woon University, Seoul, Republic of Korea, in 2013. Currently, he is working toward a M.S degree in computer engineering at the same university. His current research interests include video coding and computer vision.



Yong-Jo Ahn was born in Jeju Province, Republic of Korea, in 1984. He received his B.S. and M.S. degree in Computer Engineering from Kwang-woon University, Seoul, Korea, in 2010 and 2012, respectively. Now, he is working toward a Ph.D. degree at the same university. His current research interests include high-efficiency video compression techniques, parallel processing for video coding and multimedia system.

Jung-Soo Mok received his B.S. degree in Computer Engineering from Kwangwoon University, Seoul, Republic of Korea, in 2013. Currently, he is working towards a M.S degree in computer engineering at the same university. His current research interests include video coding and computer vision.

Donggyu Sim received his B.S. and M.S. degrees in Electronic Engineering from Sogang University, Korea, in 1993 and 1995, respectively. He also received the Ph.D. degree at the same University, in 1999. He was with Hyundai Electronics Co., Ltd. From 1999 to 2000, being involved in MPEG-7 standardization. He was a senior research engineer at Varo Vision Co., Ltd, working on MPEG-4 wireless applications from 2000 to 2002. He worked for the Image Computing Systems Lab. (ICSL) at the University of Washington as a senior research engineer from 2002 to 2005. He researched ultrasound image analysis and parametric video coding. Since 2005, he has been with the Department of Computer Engineering at Kwangwoon University, Seoul, Korea. In 2011, he joined the Simon Frasier University, as a visiting scholar. He was elevated to an IEEE Senior Member in 2004. He is one of main inventors in many essential patents licensed to MPEG-LA for HEVC standard. His current research interests include video coding, video processing, computer vision, and video communication.