Multimed Tools Appl (2017) 76:25271–25284
DOI 10.1007/s11042-017-4413-7

Prediction complexity-based HEVC parallel processing for asymmetric multicores

Hyun-Joon Roh[1] · Sung Won Han[2] · Eun-Seok Ryu[1]

Received: 18 August 2016 / Revised: 6 December 2016 / Accepted: 20 January 2017 /
Published online: 9 February 2017

Abstract This paper proposes a novel Tile allocation method considering the computational ability of asymmetric multicores as well as the computational complexity of each Tile. This paper measures the computational ability of asymmetric multicores in advance, and measures the computational complexity of each Tile by using the amount of HEVC prediction unit (PU) partitioning. The implemented system counts and sorts the amount of PU partitions of each Tile, and also allocates Tiles to asymmetric big.LITTLE cores according to their expected computational complexity. When experiments were conducted, the amount of PU partitioning and the computational complexity (decoding time) showed a close correlation, and average performance gains of decoding time with the proposed adaptive allocation were around 36 % with 12 Tiles, 28 % with 18 Tiles, and 31 % with 24 Tiles, respectively.

Keywords HEVC · Parallel processing · Asymmetric multicores · Prediction Complexity

Hyun-Joon Roh and Sung Won Han contributed equally to this work.

✉ Eun-Seok Ryu
esryu@gachon.ac.kr

Hyun-Joon Roh
ggyo@gc.gachon.ac.kr

Sung Won Han
swhan@korea.ac.kr

[1] Department of Computer Engineering, Gachon University, 1342, Seongnam-daero, Sujeong-gu, Seongnam-si, Gyeonggi-do, 13120, Republic of Korea

[2] School of Industrial Management Engineering, Korea University, 145 Anam-Ro, Seongbuk-Gu, Seoul, 02841, Republic of Korea

1 Introduction

Recently, research and development on multimedia content processing for Ultra High Definition (UHD) video is emerging. South Korea plans to provide ground wave broadcast with 4K UHD videos in 2017 while also scheduling to provide worldwide broadcasting with 4K UHD videos at Pyeongchang 2018 Winter Olympic Games. Besides, Japan aims to broadcast Tokyo 2020 Summer Olympic Games with 8K UHD videos. The 4K and 8K UHD videos have resolutions 4 times and 16 times larger than Full High Definition (FHD) video, respectively. However, the resolution of UHD makes the parallel processing to support real-time requirements very important.

To support larger UHD resolutions, the Joint Collaborative Team on Video Coding (JCT-VC) in January 2013 developed the High Efficiency Video Coding (HEVC) standard with better video compression technologies. *Tile* is a new parallel processing tool for the HEVC standard and is specially designed for the parallel processing purpose by segmenting each picture into multiple rectangular regions and allocating them to Central Processing Unit (CPU) cores.

Recently, many CPU manufacturing companies have manufactured devices with asymmetric multicores, which have features that save energy while delivering performance that is similar to symmetric core CPUs. For example, recent ARM processors incorporate the *big.LITTLE* architecture that uses asymmetric multicores. big.LITTLE allocates light-load threads into little cores (slower core) and allocates heavy-load threads into big cores (faster core) for improved energy efficiency and better performance [3, 5, 6, 9, 10, 14]. The problem with Tile-based parallel processing technology, however, is that it does not consider the computational ability of asymmetric CPU cores but rather allocates video Tiles to the cores equally, which causes video processing (encoding/decoding) delays as shown in Fig. 1. Thus, this paper proposes a novel Tile allocation method by considering the computational ability of asymmetric multicores and the computational complexity of each Tile. The computational ability of asymmetric multicores can be measured or provided by the manufacturer while the computational complexity of each Tile can be measured by the amount of HEVC prediction unit (PU) partitioning. Then, based on the measured/analyzed complexity of tiles and computing power of asymmetric cores, the proposed method allocates the set of Tiles to the Big and Little cores as shown in Fig. 2.

The paper is organized as follows. Section 2 provides several related work including HEVC standard and complexity prediction algorithms. Section 3 proposes the Tile allocation algorithm for asymmetric multicores, and describes the logic in detail with the

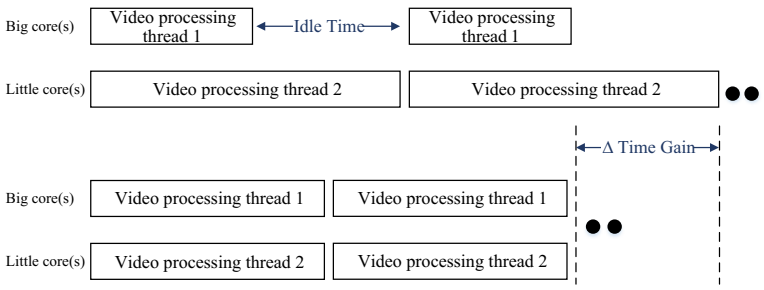


Fig. 1 Expected decoding time gain between uniform Tile allocation (*above*) and non-uniform Tile allocation (*below*)

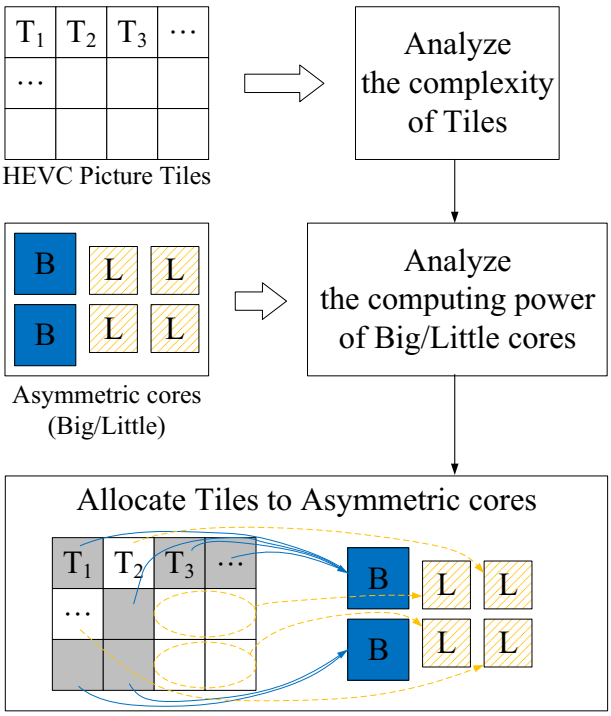


Fig. 2 Conceptual diagram of the proposed Tile allocation method

implemented modules in HEVC reference software HM. The experimental results are described in Section 4; lastly, Section 5 summarizes the conclusions of this study and outlines future work.

2 Related work

This section details the background information necessary to understand the proposed Tile allocation algorithm for asymmetric multicores.

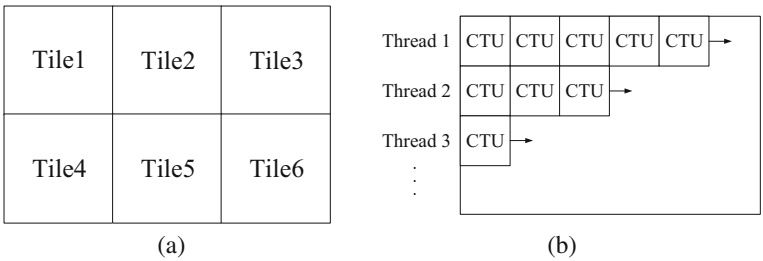


Fig. 3 Two parallel processing tools of HEVC; **a** example of a frame divided into Tiles, **b** example of WPP

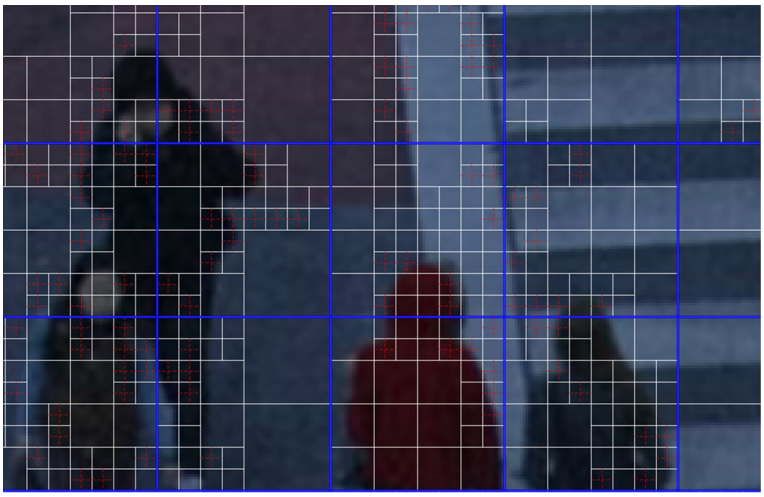


Fig. 4 An example of CU and PU partitions with *PeopleOnStreet* test sequence

2.1 JCT-VC HEVC video coding standard with Tile tool

After successfully standardizing H.264/AVC [13], ISO/IEC MPEG and ITU-T VCEG jointly organized the Joint Collaborative Team on Video Coding (JCT-VC) to develop the next generation video standard named HEVC. This new standard targets UHD displays that have 4K / 8K pixels in a horizontal line. Because of the ultra high resolution that requires very high computing power, the HEVC includes two parallel processing tools in it.

The HEVC parallel processing tools support different picture partition strategies such as Tiles and Wavefront Parallel Processing (WPP), as shown in Fig. 3. Tiles partition a picture with horizontal and vertical boundaries so that it provides better coding gains compared to multiple slices [7]. WPP is used when a slice is divided into rows of coding tree units (CTUs). In WPP, the first row of CTUs is decoded normally, but each additional row requires decisions from the previous row. WPP has the entropy encoder that uses information from the preceding row of CTUs, and allows for parallel processing, which results in better compression as compared to Tiles [8, 12].

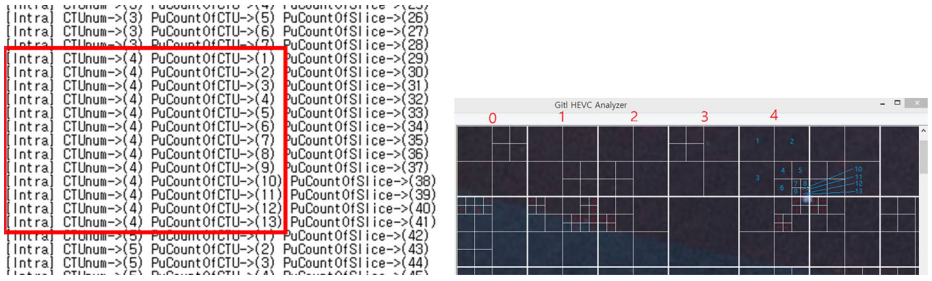


Fig. 5 Verifying the implemented PU partitions counting module; **a** the log of implemented module, **b** HEVC Analyzer

2.2 Encoding/decoding complexity prediction algorithms

Decoding complexity of video frames is influenced by many explicit factors including encoding parameters such as resolution, QP, and objects in picture. Recent researches on predicting encoding/decoding complexity are centered on optimizing power efficiency and encoding/decoding time.

One of those researches aims at adjusting clock speed of CPU to save power [1]: it sets low clock speed when a frame has less computational complexity and high clock speed for high computational complexity frames. This research is meaningful in improving power efficiency but it does not consider parallelism on multicore systems and optimization of decoding time.

Another research proposed a Tile partitioning algorithm based on the number of bits of CTUs [2]. The logic proposes a method to equalize the total number of bits in each Tiles in order to minimize the decoding time between Tiles that have a lot of bits or a few bits. It has many similarities with our research, but it uses a different method to predict complexity and does not consider asymmetric multicore environments.

3 Prediction complexity-based HEVC parallel processing for asymmetric multicores

This section introduces our proposed method, prediction complexity-based HEVC parallel processing for asymmetric multicores in detail. The concept of CU in HEVC is similar to the macro block (MB) in H.264 AVC standard. For example, one partitioning mode can split a 64×64 CU into 32×32, 16×16, and 8×8 CUs, and each CU can be partitioned into one, two, or four PUs. The PU is a basic unit for prediction in CU as shown in Fig. 4.

Because the HEVC profiling results explain a motion compensation in video coding standard, it requires significant computing power [4]. This study assumes that the amount of PU partitions of a CU has a high correlation with computational complexity. There have been

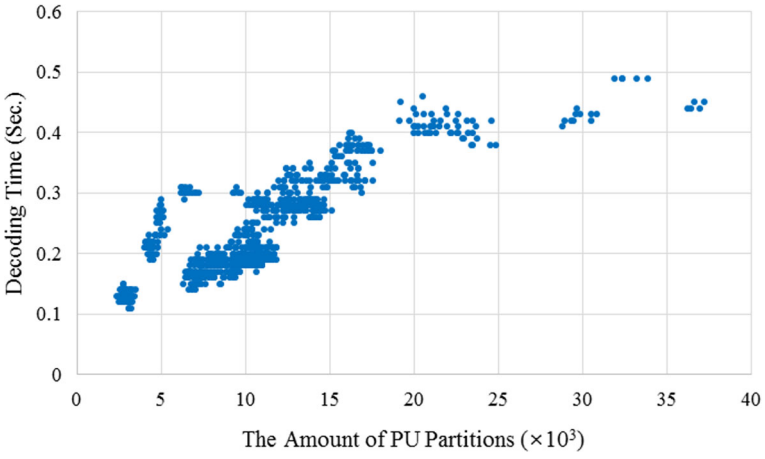


Fig. 6 High correlation between the sum of PU partitions and actural decoding time; the bitstream is encoded with RA configuration

researches to estimate picture complexity; however, the PU partition-based complexity estimation is possibly one of the simplest methods because it is measured and given information by encoder while it encodes each CU.

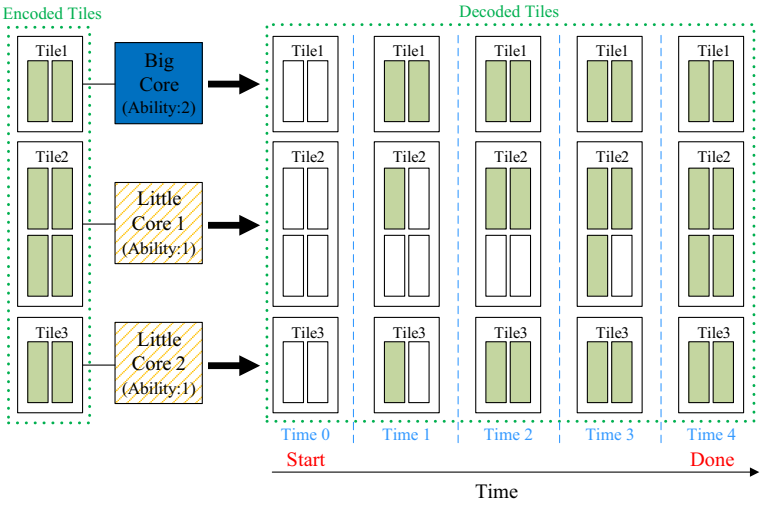


Fig. 7 Comparison of **a** the conventional Tile allocation method without considering asymmetric cores and **b** the proposed Tile allocation method with considering asymmetric cores

The modified HM encoder calculates the sum of the number of PU partitions of each CU in a Tile: if a Tile has many PU partitions, the processing complexity of the Tile is high. Therefore to verify our assumption, we implemented the counting module of PU partitions in HM reference encoder and verified the implementation with the GitlHEVCAnalyzer (version 1.5.1) visualization tool [11], as shown in Fig. 5.

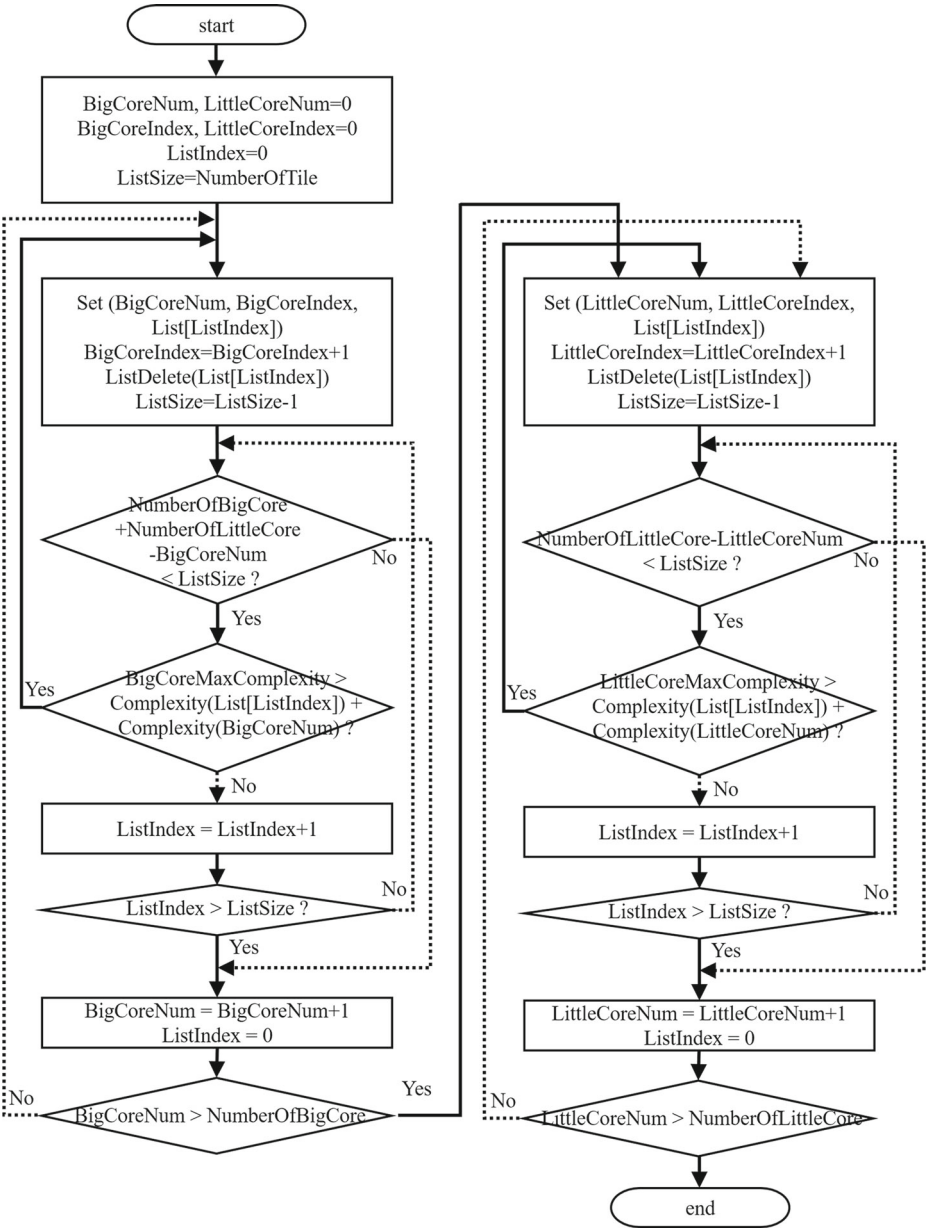


Fig. 8 Proposed Tile allocation algorithm for asymmetric multicores

Table 1 Experimental environment with commercial smart phone (Galaxy S7 edge)

Big cluster	Exynos M1
Little cluster	Cortex-A53
Number of cores	4 Big cores (2.3 GHz) 4 Little cores (1.6 GHz)
Memory	4GB DDR RAM

The actual decoding time of each Tile is then measured and compared with the amount of PU partitions. Figure 6 shows the relationship between the sum of PU partitions and actual decoding time. In the result, Y-axis denotes the decoding time, and the X-axis represents the amount of PU partitions. The plot of bitstreams encoded with RA configuration shows high correlation between the amount of PU partitions and decoding time. Moreover, this paper uses the bitstreams with RA configuration defined in HEVC common test condition (CTC), wherein the seven test sequences (two 4K UHD videos, five HD videos) are used. This paper also set the quantization parameters (QP) as 22, 27, 32, and 37.

3.1 Tile allocation algorithm for asymmetric multicores

Figure 7 explains the example of the problem of the conventional Tile allocation without considering asymmetric cores. In the example, a gray colored rectangular block implies a job unit that requires a certain amount of processing time. That is, if *Tile 2* in the figure has 4 job units, it means the computational complexity of the *Tile 2* is twice of the *Tile 3*. Thus, in the example, the proposed method saves two times of the total decoding time compared to the conventional method (conventional: takes time 4, proposed: takes time 2).

Under the assumption that complexity is positively correlated to computational time, we attempt to minimize the maximum of complexities assigned to cores. The model formulation is as follows: Let the C_k be the complexity assigned to the core k, where $k = 1, 2, ..., K$, and K is the number of cores. Let P_i be the PU of the i-th Tile, and S_k be the set of Tiles that are assigned to core k. The optimization problem can be formulated as

$$\min C^* \tag{1}$$

such that

$$C^* = \max_k \{C_k\}$$

$$\sum_{i \in S_k} P_i = C_k$$

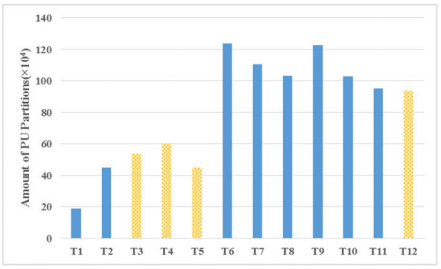
Table 2 Test sequences with class A+ and B in CTC

Name	Resolution	Frame length	Frame rate
PeopleOnStreet	3840 × 2160	150	30 fps
Traffic	3840 × 2048	300	30 fps
BasketballDrive	1920 × 1080	500	50 fps
BQTerrace	1920 × 1080	600	60 fps
Cactus	1920 × 1080	500	50 fps
Kimono	1920 × 1080	240	24 fps
ParkScene	1920 × 1080	240	24 fps

(a) (b) (c)

Fig. 9 Test sequence *PeopleOnStreet* with **a** 12 Tiles, **b** 18Tiles and **c** 24 Tiles

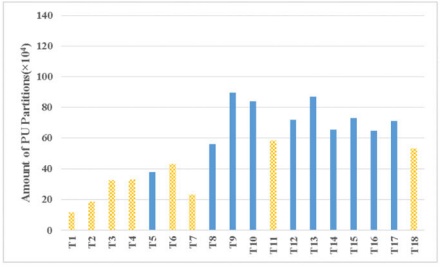
The objective of the optimization problem is to minimize the maximum complexity among the cores. For the problem, we proposed a heuristic algorithm that finds a better solution than the conventional approach. The algorithm we proposed is illustrated in Fig. 8 below:



L1	L2	L3	L4
B1	B2	L1	L2
L3	L4	B1	B2

B1	B2	L3	L2
L4	B1	B1	B2
B1	B2	B2	L1

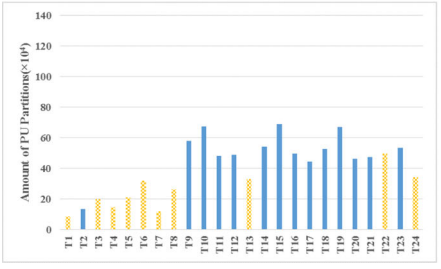
(a) (b) (c)



L1	L2	L3	L4	B1	B2
L1	L2	L3	L4	B1	B2
L1	L2	L3	L4	B1	B2

L2	L1	L4	L4	B1	L3
L3	B2	B1	B1	L1	B1
B1	B2	B2	B2	B2	L2

(d) (e) (f)



L1	L2	L3	L4	B1	B2
L1	L2	L3	L4	B1	B2
L1	L2	L3	L4	B1	B2
L1	L2	L3	L4	B1	B2

L4	B1	L3	L4	L3	L2
L4	L2	B1	B1	B2	B2
L1	B1	B1	B2	B2	B2
B1	B2	B2	L3	B1	L1

(g) (h) (i)

Fig. 10 Tile allocation map of a picture with 12, 18 and 24 Tiles (B: a Tile allocated to a big core, L: a Tile allocated to a little core)

Figure 8 shows the proposed Tile allocation algorithm for asymmetric multicores. Using Tiles list with their complexity list, the algorithm allocates the Tiles with high complexity to big cores first, and then the Tiles with low complexity to little cores.

4 Experimental results

This experiment, described in Table 1, uses HM15.0 (HEVC reference model) to encode test sequences and decode them with implemented HEVC decoder on Galaxy S7 edge that

Table 3 The gains of decoding time speed up using uniform Tile allocation and proposed adaptive Tile allocation

Test sequence	QP	Decoding time gain (%)		
		Tile 12	Tile 18	Tile 24
PeopleOnStreet	22	31.43	22.67	14.97
	27	32.08	22.77	17.13
	32	32.75	22.84	17.21
	37	33.49	22.36	20.09
Traffic	22	31.63	26.41	28.49
	27	34.67	27.27	30.48
	32	37.37	28.55	31.33
	37	38.26	29.19	31.95
BasketballDrive	22	37.13	27.06	34.84
	27	37.93	28.95	35.74
	32	38.82	29.01	33.85
	37	39.63	29.19	33.17
BQTerrace	22	35.89	29.66	37.77
	27	36.72	27.54	36.25
	32	36.55	26.02	35.48
	37	35.40	25.36	36.27
Cactus	22	38.27	39.82	36.78
	27	38.64	39.84	36.14
	32	39.81	38.22	32.80
	37	38.63	37.62	32.11
Kimono1	22	34.82	27.27	31.23
	27	34.67	24.59	30.63
	32	35.14	25.66	28.13
	37	35.23	24.09	30.40
ParkScene	22	34.58	30.42	34.04
	27	35.65	28.29	34.89
	32	37.20	28.17	34.76
	37	36.85	31.06	35.69
Average		36.04	28.57	31.17

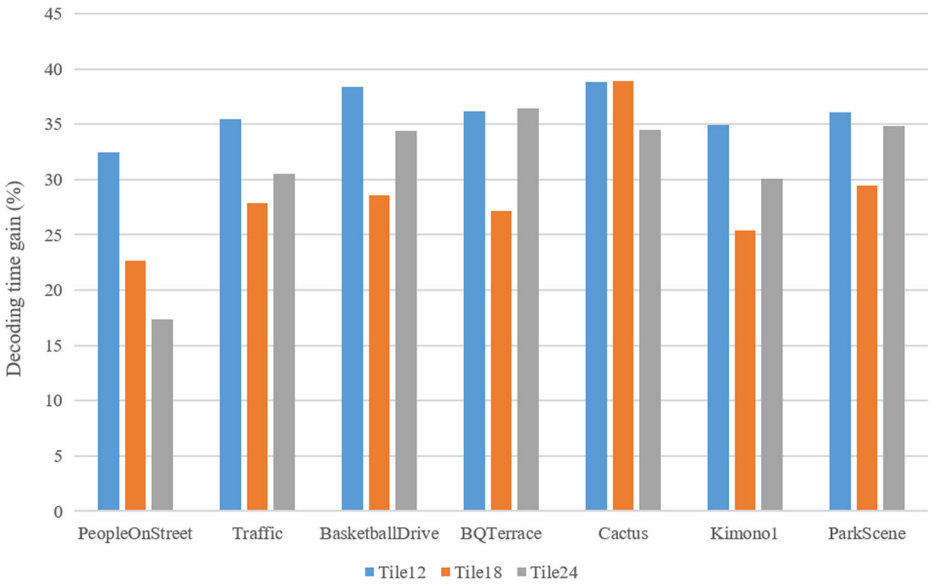


Fig. 11 Decoding time speed up with proposed Tile allocation method

supports asymmetric big.LITTLE multicores. For the encoding and decoding, seven test sequences described in Table 2, RA configuration, and four QP values (22, 27, 32, 37) are applied. The video picture is divided into 12, 18, and 24 Tiles as shown in Fig. 9. For the practical real experiments, this experiments are conducted on the latest smart phone, Galaxy S7 edge, that has 8 cores including 4 big cores and 4 little cores. In the experiment, this paper uses 2 big cores and 4 little cores because the phone sets 2 big cores as idle state for saving battery. Thus, the video picture is divided into multiple Tiles, and the number of Tiles has multiple of 6 (number of available cores in normal state). According to the simple pilot test, we checked that the big core is 3 times faster than the little core for decoding with the phone. Thus, the proposed algorithm tries to make total complexity of Tiles that allocated to big core has three times of the one to little core.

Figure 10 shows the measured decoding times of Tiles ((a) Tile 12, (d) Tile 18, (g) Tile 24) at *PeopleOnStreet*. As shown in Fig. 10, f and i, the proposed method allocates Tiles with high complexity to big cores, and the decoding speed up gain were around 32 %, 22 %, 17 %.

Table 3 and Fig. 11 shows the gains from conventional and proposed methods. When the performance was measured with Tile 12, Tile 18, and Tile 24, the gains with allocating number of Tiles to cores adaptively (different number of Tiles) were around 30 %.

5 Conclusion

This paper proposes a novel Tile allocation method for parallel HEVC processing with consideration for the computational ability of asymmetric multicores as well as the

computational complexity of each Tile. The computational ability of asymmetric multicores can be measured or provided by chip maker, and the computational complexity of each Tile can be measured by the amount of HEVC prediction unit (PU) partitioning.

The implemented system (i) counts and sorts the amount of PU partitioning of each Tile and (ii) allocates Tiles to asymmetric big.LITTLE cores according to their expected computational complexity. 4K UHD and HD test sequences with RA coding structure, and 6 asymmetric multicores (2 big cores and 4 little cores) were used for experiments. When the experiments were conducted, the amount of PU partitioning and the computational complexity (decoding time) showed close correlation, and average performance gains of decoding time were around 36 % for 12 Tiles, 28 % for 18 Tiles, and 31 % for 24 Tiles, respectively.

Acknowledgments 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-2015R1C1A1A02037743).

References

1. Ahn H, Jeong S (2013) Power-minimizing dvfs algorithm using estimation of video frame decoding complexity. J Kor Inst Commun Inf Sci 38(1):46–53
2. Baik H, Song H (2015) A complexity-based adaptive tile partitioning algorithm for HEVC decoder parallelization. In: 2015 IEEE international conference on image processing (ICIP). IEEE, pp 4298–4302
3. Blem E, Menon J, Sankaralingam K (2013) Power struggles: Revisiting the risc vs. cisc debate on contemporary arm and x86 architectures. In: 2013 IEEE 19th international symposium on high performance computer architecture (HPCA2013). IEEE, pp 1–12
4. Bossen F, Bross B, Suhring K, Flynn D (2012) HEVC complexity and implementation analysis. IEEE Trans Circuits Syst Video Technol 22(12):1685–1696
5. Carroll A, Heiser G (2014) Unifying dvfs and offlining in mobile multicores. In: 2014 IEEE 19th real-time and embedded technology and applications symposium (RTAS). IEEE, pp 287–296
6. Koufaty D, Reddy D, Hahn S (2010) Bias scheduling in heterogeneous multi-core architectures. In: Proceedings of the 5th European conference on Computer systems. ACM, pp 125–138
7. Misra K, Segall A, Horowitz M, Xu S, Fuldseth A, Zhou M (2013) An overview of tiles in HEVC. IEEE J Sel Top Sign Proces 7(6):969–977
8. Ohm JR, Sullivan GJ, Schwarz H, Tan TK, Wiegand T (2012) Comparison of the coding efficiency of video coding standards–including high efficiency video coding (HEVC). IEEE Trans Circuits Syst Video Technol 22(12):1669–1684
9. Pricopi M, Muthukaruppan TS, Venkataramani V, Mitra T, Vishin S (2013) Power-performance modeling on asymmetric multi-cores. In: 2013 international conference on compilers, architecture and synthesis for embedded systems (CASES). IEEE, pp 1–10
10. Shelepov D, Saez Alcaide JC, Jeffery S, Fedorova A, Perez N, Huang ZF, Blagodurov S, Kumar V (2009) Hass: a scheduler for heterogeneous multicore systems. ACM SIGOPS Operating Syst Rev 43(2):66–75
11. Springer D, Schnurrer W, Weinlich A, Heindel A, Seiler J, Kaup A (2014) Open source HEVC analyzer for rapid prototyping (harp). In: 2014 IEEE international conference on image processing (ICIP). IEEE, pp 2189–2191
12. Sullivan GJ, Ohm JR, Han WJ, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol 22(12):1649–1668
13. Sullivan GJ, Topiwala PN, Luthra A (2004) The H. 264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions. In: Optical Science and Technology, the SPIE 49th Annual Meeting. International Society for Optics and Photonics, pp 454–474
14. Yoo S, Shim Y, Lee S, Lee SA, Kim J (2015) A case for bad big. little switching: How to scale power-performance in si-hmp. In: Proceedings of the workshop on power-aware computing and systems. ACM, pp 1–5

Hyun-Joon Roh is a Master student at the department of Computer Engineering in Gachon University, Seongnam, Korea. He received his B.S. in computer engineering from Gachon University in 2016. His research focuses on optimizing parallel video processing using asymmetric multicore processors. He is a Student Member of the IEEE, IEEE Computer Society.

Sung Won Han is an assistant professor in the School of Industrial Management Engineering, Korea University. Previously, he was a senior research scientist at the division of Biostatistics, School of Medicine, New York University, and was a post-doctoral researcher in the department of Biostatistics and Epidemiology/Center for Clinical Epidemiology and Biostatistics, School of Medicine, University of Pennsylvania. he received a Ph.D. in the School of Industrial and Systems Engineering (Statistics) at the Georgia Institute of Technology, where he also received an M.S. in operations research in 2006, an M.S. in statistics in 2007, and an M.S. in mathematics in 2010. His research focuses on statistical methodologies and data analytics, which cover spatiotemporal surveillance for chronic diseases, functional change-point problems, and gene network problems in high dimensional data.

Eun-Seok Ryu is an Assistant Professor at the Department of Computer Engineering in Gachon University, Seongnam, Korea. Prior to joining the University in 2015, he was a Principal Engineer at Samsung Electronics, Suwon, Korea, where he led a multimedia team. He was a Staff Engineer at InterDigital Labs, San Diego, California, USA, from Jan. 2011 to Feb. 2014, where he researched and contributed to next generation video coding standards such as HEVC and SHVC. From Sep. 2008 to Dec. 2010, he was a Postdoctoral Research Fellow at GCATT in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia USA. In 2008, he was a Research Professor at the research institute for information and communication technology in Korea University, Seoul, Korea. His research interests are in the area of multimedia communications that includes video source coding and wireless mobile systems. He received his B.S., M.S., and Ph.D. in computer science from Korea University in 1999, 2001, and 2008, respectively. He is a Senior Member of the IEEE, IEEE Computer Society, and IEEE Communications Society.