IEIE Transactions on Smart Processing and Computing, vol. 5, no. 4, August 2016
http://dx.doi.org/10.5573/IEIESPC.2016.5.4.235

IEIE Transactions on Smart Processing and Computing

Operation-level Early Termination Algorithm for Inter-predictions in HEVC

Chae Eun Rhee

Department of Information and Communication, Inha University / Incheon, South Korea chae.rhee@inha.ac.kr

Received June 27, 2016; Accepted July 21, 2016; Published August 30, 2016

* *Regular Paper*

***Abstract*:** The emerging High-Efficiency Video Coding (HEVC) standard attempts to improve coding efficiency by a factor of two over H.264/Advanced Video Coding (AVC) at the expense of an increase in computational complexity. Mode decision with motion estimation (ME) is still one of the most time-consuming computations in HEVC, as it is with H.264/AVC. Thus, fast mode decisions are not only an important issue to be researched, but also an urgent one. Several schemes for fast mode decisions have been presented in reference software and in other studies. However, the conventional hierarchical mode decision can be useless when block-level parallelism is exploited. This paper proposes operation-level exploration that offers more chances for early termination. An early termination condition is checked between integer and fractional MEs and between the parts of one partition type. The fast decision points of the proposed algorithm do not overlap those in previous works. Thus, the proposed algorithms are easily used with other fast algorithms, and consequently, independent speed-up is possible.

***Keywords*:** HEVC, Inter-prediction, Fast mode decision

1. Introduction

H.264/Advanced Video Coding (AVC) [1] was regarded as the state-of-the-art video coding standard and is still widely used for various video applications due to the increasing demand for high-definition (HD) video content. Recently, the next-generation video coding standard [2-4] known as High-Efficiency Video Coding (HEVC) was developed by the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG). In the emerging HEVC standard, several new features are introduced, including a coding tree unit (CTU), intra-coding with increased spatial prediction directions, more sophisticated interpolation filters, various in-loop filters, and enhanced entropy coding schemes. In particular, various block sizes, from 8×8 to 64×64, and flexible block size selection schemes for predictions and transforms have been shown to be very suitable for large resolutions. Large resolutions such as 4k×2k and beyond (up to 8k×4k) are challenging targets for new video applications and consumer devices. The HEVC standard aims to double the bitrate reduction offered by H.264/AVC at the expense of an increase in computational complexity. Mode decisions with motion estimation (ME) are among the most time-consuming computations in HEVC, as they are with H.264/AVC. Thus, fast inter-prediction is not only an important issue to be researched but also an urgent one.

Extensive research efforts have been conducted to reduce the inter-prediction computational complexity for H.264/AVC. There are roughly three categories of algorithms for fast inter-mode predictions. In the first category, candidate block sizes are determined prior to ME, and prediction operations including ME are performed for only the selected candidate block sizes. For this pre-decision, spatial and/or temporal correlation from neighboring information, motion, or texture characteristics are explored [5-15]. In the second category, the results of the prior predictions are compared and further predictions are determined according to the comparison result. Mode decisions are made in a hierarchical manner [13-20]. In the third category, the fractional ME (FME) calculation is reduced using integer ME (IME) results. Candidate block sizes for FME are selected from the IME results. Or FME computations are reused for other blocks that have an identical integer motion vector (MV) [21-23]. However, these solutions cannot be directly applied to HEVC encoding due to the new coding structures and different data processing procedures, compared to H.264/AVC. HEVC reference software and other recent studies have suggested several schemes for rapid mode decisions. These algorithms are useful for speed-up in most cases. However, the conventional fast mode decision, which is made in a hierarchical manner, can be useless when block-level parallelism is exploited in the hardware implementation or multi-core environment.

In this paper, operation-level exploration is conducted to realize more chances for early termination. In this algorithm, inter-prediction for a particular block partition is divided into several operations. IME and FME should be performed sequentially in ME. Between IME and FME, the early termination condition is checked. If this condition is met, FME for the current block partition is skipped, and the search moves on to other block partitions. In addition, some block partition types consist of several parts in which MEs are executed for each part independently. Between parts in a partition type, a similar early termination scheme is applied. The proposed operation-level early termination algorithm is still effective in block-level parallel execution. The fast decision points of the proposed algorithms do not overlap those proposed in the HEVC reference software or in previous studies. Thus, the proposed algorithms are easily used with other fast-decision algorithms, and the resulting speed-up effect is quite independent. Simulation results show that the encoding speed is improved by 16%, whereas the bitrate increases by less than 0.89% with a minor peak signal-to-noise ratio (PSNR) degradation at an average of 0.028dB. Compared to previous works, similar speed-up results are guaranteed for various test sequences that have different motion and texture characteristics. When the proposed algorithm is used with the fast schemes presently available in the HEVC reference software, the time saving is 36%, whereas the increase in the bitrate and the PSNR drop are 1.19% and 0.036dB, respectively. The rest of the paper is organized as follows. Section 2 gives an overview and analysis of inter-prediction in HEVC. The proposed algorithm and simulation results are presented in sections 3 and 4, respectively. Conclusions are given in Section 5.

2. Overview and Analysis of Fast Inter-prediction in HEVC

In the existing video standards, the macroblock (MB) has served as the basic processing unit for a long time. The size of an MB is 16×16 pixels in terms of the luma component. To achieve high compression performance for high-resolution videos, instead of the MB, HEVC defines the coding unit (CU) as the basic unit. Unlike an MB, the size of a CU is not fixed, varying from 8×8 to 64×64. Given the CU size, a variable block type of quadtree structure is adopted. The largest CU at depth 0 is denoted by LCU. When the tree depth is 4, the size of the LCU is 64×64. The LCU can be split into as many as four 32×32 CUs. Each 32×32 CU can be split further into four 16×16 CUs. Assuming that the size of a particular CU is 2N×2N, a CU can be split into 2N×2N, 2N×N, and N×2N types of prediction units (PUs). Inter- and intra-predictions are performed for each PU.

HEVC supports variable block sizes of a quadtree structure, and the depth of this tree can be as large as 4. Two hundred fifty-five block partitions in total need to be searched. Thus, the accurate selection of candidate block sizes before ME is very difficult. In contrast, a hierarchical decision algorithm is very effective for HEVC because there are many opportunities to terminate further predictions while searching a tree of CUs. For a hierarchical block size decision, the reference software includes several fast-decision algorithms. For an early SKIP mode decision at every depth, the ME for a 2N×2N PU and the computation for the SKIP mode are performed sequentially. If the SKIP cost is less than the ME cost of the 2N×2N PU, the early skip condition is satisfied. Or, if the ME cost of the 2N×2N PU is less than the SKIP cost and both the coded block flag (CBF) and the MV difference (MVD) of the 2N×2N PU are zero, the early skip condition is also satisfied. The CBF represents blocks with a zero residual. If the early skip condition is true, the remaining inter- and intra-predictions in the current CU are not searched. This scheme is called the early SKIP mode detection (ESD) scheme. Other schemes for fast inter-predictions are the early CU determination and the CBF-based fast mode schemes, denoted as ECU and CFM, respectively. In the ECU scheme, if the SKIP mode is the best mode at the current CU, the predictions for the smaller CUs at the next depth are not performed. For example, if the SKIP mode cost of a 64×64 PU in a 64×64 CU is the smallest among 64×64, 64×32, and 32×64 PUs, the block size search is terminated at this depth, and it becomes unnecessary to run the predictions for the smaller CUs (i.e., the 32×32, 16×16, and 8×8 CUs). Meanwhile, the CFM scheme is used to select the PU size early in the current CU and to reduce the amount of computation needed for predictions with less-probable PU sizes. The predictions for the SKIP, 2N×2N, 2N×N, and N×2N PUs are processed one by one. During these predictions, if the CBF of the current PU happens to be all zeros, the prediction for the current CU is terminated, and the computation for the remaining PU sizes is thus reduced, as a zero CBF indicates that the rate-distortion (R-D) performance is adequate when the current PU is determined to be the best mode. Even if the current PU is different from the best PU, the difference in the R-D cost between the current PU and the best PU may be negligible.

The ECU saves a lot of search time through the early mode decision of the best CU. The goal of the ECU is not to search small CUs. To this end, the best mode of the current CU should be the SKIP mode. However, the ECU is not useful for videos that have an inhomogeneous texture or complex motion. After encoding these types of videos, it is observed that many small CUs are included, and their coefficients are quite large. Fig. 1 shows the probability of the ECU condition being satisfied. The light gray part represents the rate of CUs where the best mode is not the SKIP mode, whereas the dark gray part represents the rate of CUs for which the best mode is the SKIP mode. In Figs. 1(a) and (b), KristenAndSara at a resolution of 1280×720 is used, whereas RaceHorses at a resolution of 832×480 is used in Figs. 1(c) and (d). Two different quantization parameter (QP) values of 20 and 36 are

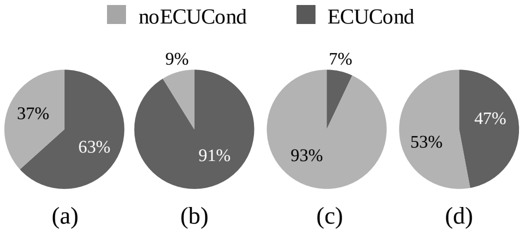

Fig. 1. The rate of the ECU (a) KristenAndSara when QP=20, (b) KristenAndSara when QP=36, (c) RaceHorses when QP=20, (d) RaceHorses when QP=36.

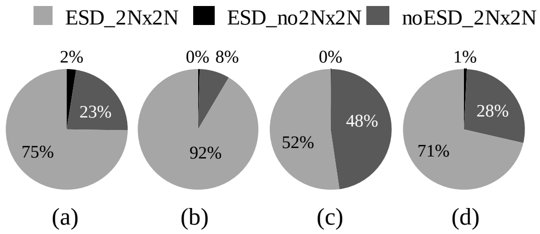

Fig. 2. The rate of the ECU (a) KristenAndSara when QP=20, (b) KristenAndSara when QP=36, (c) RaceHorses when QP=20, (d) RaceHorses when QP=3636.

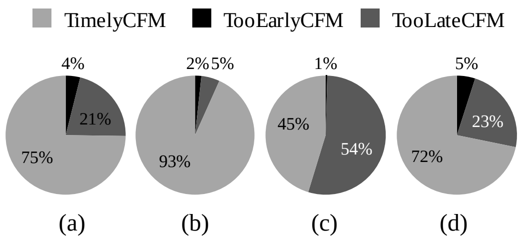

Fig. 3. The rate of the ECU (a) KristenAndSara when QP=20, (b) KristenAndSara when QP=36, (c) RaceHorses when QP=20, (d) RaceHorses when QP=36.

applied to each video. In the test sequence KristenAndSara, the motion changes smoothly, and the texture is also homogeneous. Thus, a large number of CUs satisfy the ECU condition. With a QP of 36, 91% of the CUs are determined as SKIP mode. In contrast, the RaceHorses test sequence has complex motion and spatial details. With a QP equal to 20, only 7% of the CUs satisfy the ECU condition. This ECU scheme is effective only for videos that have spatially and temporally homogeneous characteristics or that are encoded with a high QP value.

ESD and CFM are used to determine the best PU at the current depth. In Figs. 2 and 3, the suitability of the ESD and CFM algorithms is analyzed. KristenAndSara and RaceHorses are encoded with QPs of 20 and 36, respectively. In Fig. 2, ESD_2N×2N (denoted in light gray) indicates that the ESD condition is satisfied and that the best PU is determined as SKIP or 2N×2N inter-modes in the end. In this case, when the ESD is applied, a time reduction is achieved without any loss in quality. Meanwhile, ESD_no2N×2N (denoted in black) represents an incorrect decision in which the best PU is neither SKIP nor 2N×2N inter-modes, even if the ESD condition is satisfied. This portion causes degradation in the R-D performance. Lastly, noESD_2N×2N (shown in dark gray) represents the portion of PUs not covered by the ESD scheme, where the best mode is from among the SKIP and 2N×2N inter-modes, even if the ESD condition is not satisfied. Similar to the results of Fig. 1, the dark gray portion increases with the complexity of the videos or as QP values become smaller. This uncovered portion should be exploited to save more time. The CFM scheme in Fig. 3 shows a trend very similar to ESD in Fig. 2. Here, timelyCFM (denoted in light gray) indicates that early termination using CFM is properly applied immediately after the best PU size. The part shown in dark gray represents the TooLateCFM decision. Here, the current PU, which was just tested, is the best one in the end. However, the PU search should continue because the CBF of the current PU is not zero. If no PU has a zero CBF, the CFM scheme is of no use in reducing the required time for encoding. The portion of the timelyCFM decreases, and the portion of the TooLateCFM increases when videos are complex and their QP values are low, as shown in Fig. 3(c). In both Figs. 2 and 3, the portion shown in black, which causes degradation of compression efficiency, is very low. From these observations, the limits of both schemes are as follows. First, opportunities to terminate searching PUs early are too few. When the asymmetric motion partition (AMP) option is not used, the ESD condition is checked once after every CU, whereas the CFM condition is checked three times per CU after inter-predictions for 2N×2N, N×2N, and 2N×N PUs. There are only four chances in total for early termination. Second, a coefficient-based approach is not useful for videos in which the portion of blocks that have zero coefficients is not large.

3. Operation-level exploration for early termination of the PU search

To increase the possibility of PU early termination, the sum of the absolute transformed difference (SATD) costs are calculated and compared in terms of various points during the inter-predictions. To check the results of IME, FME, and each part of a PU, the SATD cost is used, as computation of the R-D cost for increased decision points is too expensive. In Fig. 4, KristenAndSara and RaceHorses are encoded with QPs of 20 and 36, respectively. The light gray bar graphs denoted as Diff_PUs represent the differences in the SATD costs among the 2N×2N, N×2N, and 2N×N PUs. The dark gray bar graphs denoted as Diff_I/FME represent the differences in the SATD costs between the IME and FME operations of the PUs, whereas the black bar graphs denoted as Diff_Parts represent the differences in the SATD costs of the two parts in the N×2N and 2N×N PUs. As shown in Fig. 4, Diff_I/FME and Diff_Parts are very small compared to Diff_PUs. This simulation result indicates that the FME cost of a PU is quite predictable from the IME cost. Likewise, the cost of the second parts of the N×2N and 2N×N PUs can be estimated from the first part.

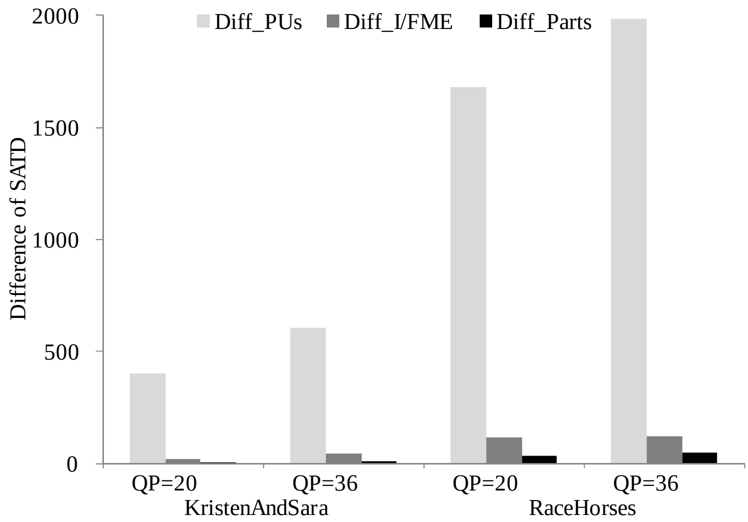

Fig. 4. Comparison of SATD cost differences.

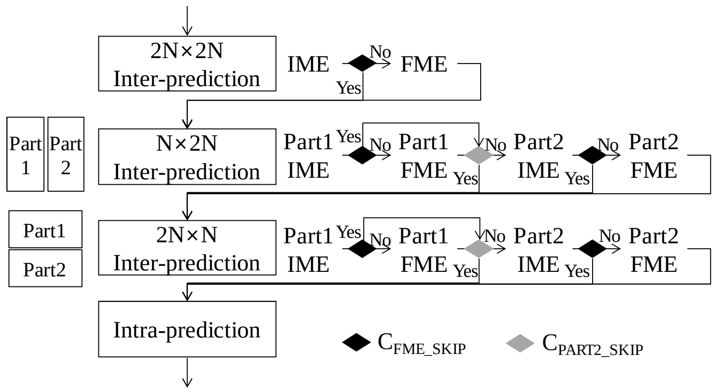

Fig. 5. The proposed operation-level early termination algorithm.

Considering Fig. 4, an operation-level exploration is proposed for the PU-level fast decision to ensure sufficient opportunities for early termination. In Fig. 5, the fine operations of the inter-predictions are described. The 2N×2N SKIP computation, which does not require the ME, is not shown. For the 2N×2N inter-prediction, the IME and FME operations are performed sequentially for a 2N×2N PU. In the N×2N inter-prediction, Part1 IME and FME operations are for one N×2N block, whereas Part2 IME and FME operations are included for the other N×2N block. The 2N×N inter-prediction is performed in a manner similar to that used for N×2N inter-prediction. The proposed fast algorithm attempts to find chances for early termination during every fine operation by checking two skip conditions. One is an FME skip condition and the other is a Part2 skip condition. An FME skip condition is for the 2N×2N, N×2N, and 2N×N inter-predictions. After IME, the IME cost is compared to the current best cost using the condition CFME_SKIP, as defined in Eq. (1). Here, COSTBEST is the SATD cost of the best mode thus far, whereas COSTIME denotes the SATD cost from the IME of the current PU. If COSTBEST is less than COSTIME multiplied by WFME_SKIP, FME is not performed for the current PU. The weight value, WFME_SKIP, is chosen experimentally and is set to 0.55 and 0.8 for the 2N×2N PU and for each part of the N×2N and 2N×N PUs, respectively. Therefore, the final ME cost of the current PU can be estimated as WFME_SKIP×COSTIME, and this estimated ME cost is compared with COSTBEST. A Part2 skip condition, denoted as CPART2_SKIP in Eq. (2), is used for N×2N and 2N×N PUs and is similar to CFME_SKIP. COSTPART1 denotes the SATD cost of the first part in the current PU. If COSTBEST is less than COSTPART1 multiplied by WPART2_SKIP, IME and FME for the second part are not performed. The weight value, WPART2_SKIP, is set to 0.8.

$$C_{FME\_SKIP}: COST_{BEST} < W_{FME\_SKIP} \times COST_{IME} \quad (1)$$

$$C_{PART2\_SKIP}: COST_{BEST} < W_{PART2\_SKIP} \times COST_{PART1} \quad (2)$$

4. Simulation Results

The proposed algorithm was implemented in HM8.2 reference software and simulated on a server with an Intel Core2 processor at 3GHz with 8GB of DDR2 RAM. For the simulation, configurations for the encoding are low-complexity, low-delay, and generalized P and B picture

Table 1. R-D performance and time savings of the proposed FME- and part-skip algorithms.

Size	Videos	FME skip			Part skip		
		BDBR (%)	BDPSNR (dB)	TS (%)	BDBR (%)	BDPSNR (dB)	TS (%)
Class A	NebutaFestival	0.28	-0.015	8.40	0.27	-0.016	5.42
	PeopleOnStreet	0.31	-0.011	13.17	0.29	-0.010	6.38
	SteamLocomotiveTrain	0.21	-0.007	13.18	0.33	-0.011	7.05
	Traffic	0.28	-0.009	16.76	0.18	-0.006	7.57
Class B	BasketballDrive	0.28	-0.006	10.59	0.42	-0.008	6.55
	BQTerrace	0.72	-0.028	13.45	0.25	-0.011	5.87
	Kimono1	0.41	-0.010	9.25	0.44	-0.010	5.39
	ParkScene	0.34	-0.011	12.62	0.18	-0.005	6.36
Class C	BasketballDrill	0.58	-0.019	10.03	0.36	-0.011	5.17
	BQMall	0.42	-0.013	13.72	0.33	-0.010	6.56
	PartyScene	0.37	-0.016	8.05	0.13	-0.006	3.66
	RaceHorses	0.35	-0.014	10.84	0.27	-0.011	5.76
Class D	BasketballPass	0.63	-0.023	14.49	0.30	-0.010	6.60
	BlowingBubbles	0.41	-0.015	11.88	0.47	-0.018	5.33
	BQSquare	0.73	-0.031	14.28	0.19	-0.007	6.48
	Mobisode2	-0.83	0.019	16.47	0.09	-0.002	7.15
Class E	FourPeople	0.14	-0.003	15.47	0.15	-0.003	7.92
	Johnny	0.48	-0.006	14.25	0.95	-0.018	7.17
	KristenAndSara	0.07	-0.003	14.37	0.49	-0.011	6.95
	Vidyo1	0.39	-0.009	16.35	0.38	-0.010	8.07
Average		0.33	-0.011	12.88	0.32	-0.010	6.37

Table 2. R-D performance and time savings of all proposed algorithms.

Size	Videos	All proposed algorithms			HM fast algorithms			Proposed algorithms+HM		
		BDBR (%)	BDPSNR (dB)	TS (%)	BDBR (%)	BDPSNR (dB)	TS (%)	BDBR (%)	BDPSNR (dB)	TS (%)
Class A	NebutaFestival	0.79	-0.041	20.85	0.06	-0.004	3.36	0.26	-0.016	19.24
	PeopleOnStreet	0.89	-0.031	29.10	0.67	-0.025	13.19	1.03	-0.038	35.36
	SteamLocomotiveTrain	0.46	-0.017	22.64	0.35	-0.010	14.64	0.48	-0.015	35.46
	Traffic	0.77	-0.023	25.84	1.29	-0.040	22.16	1.72	-0.054	46.03
Class B	BasketballDrive	0.97	-0.018	19.92	0.90	-0.016	15.20	1.16	-0.022	32.88
	BQTerrace	1.09	-0.047	21.54	1.18	-0.044	17.01	1.45	-0.056	36.06
	Kimono1	1.10	-0.026	18.97	0.60	-0.014	14.27	0.88	-0.021	30.22
	ParkScene	0.65	-0.020	21.36	1.41	-0.041	18.91	1.84	-0.054	38.25
Class C	BasketballDrill	0.91	-0.030	14.94	0.63	-0.021	14.35	1.13	-0.037	29.93
	BQMall	0.90	-0.028	21.16	0.83	-0.026	16.93	1.12	-0.034	36.22
	PartyScene	0.67	-0.030	12.27	0.38	-0.017	5.54	0.54	-0.025	17.47
	RaceHorses	0.88	-0.036	23.38	0.41	-0.018	9.60	0.64	-0.027	27.33
Class D	BasketballPass	0.97	-0.036	19.96	1.18	-0.042	23.97	1.55	-0.055	43.10
	BlowingBubbles	0.85	-0.032	16.97	1.34	-0.050	14.88	1.69	-0.064	31.51
	BQSquare	0.88	-0.038	19.77	0.62	-0.026	16.04	0.69	-0.030	34.29
	Mobisode2	0.99	-0.027	22.52	0.79	-0.024	22.40	1.05	-0.031	44.31
Class E	FourPeople	0.46	-0.013	23.29	0.32	-0.010	24.46	0.79	-0.021	47.29
	Johnny	1.67	-0.027	21.62	1.90	-0.035	26.65	2.81	-0.054	47.65
	KristenAndSara	0.74	-0.017	21.29	0.89	-0.023	23.73	1.74	-0.041	44.24
	Vidyo1	1.09	-0.030	24.21	0.65	-0.016	22.37	1.29	-0.034	45.86
Average		0.89	-0.028	21.08	0.82	-0.025	16.98	1.19	-0.036	36.13

(GPB). There is one reference frame. The fast encoding (FEN) flag is turned on. Twenty video sequences, NebutaFestival, PeopleOnStreet, SteamLocomotiveTrain, and Traffic in Class A at a resolution of 2560×1600; BasketballDrive, BQTerrace, Kimono1, and ParkScene in Class B at a resolution of 1920×1080; BasketballDrill, BQMall, PartyScene, and RaceHorses in Class C at a resolution of 832×480; BasketballPass, BlowingBubbles, BQSquare, and Mobisode2 in Class D at a resolution of 416×240; and FourPeople, Johnny, KristenAndSara, and Vidyo1 in Class E at a resolution of 1280×720, are used in the simulation. Each test sequence consists of 50 frames and is encoded with four QPs (20, 24, 28 and 32).

The proposed algorithm was applied to the HM8.2

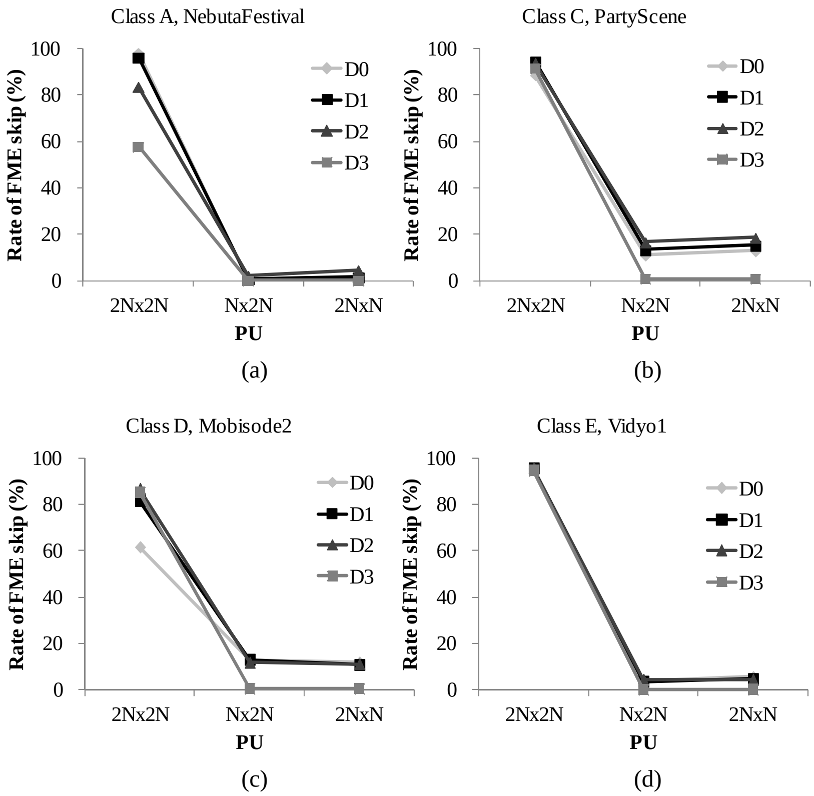

Fig. 6. The rate of PU in which FME is skipped (a) NebutaFestival in Class A, (b) PartyScene in Class C, (c) Mobisode2 in Class D, (d) Vidyo1 in Class E.

reference software, and the tabulated results are presented in Table 1. The first and second columns represent the video sizes and test sequences. From the third to the fifth columns, the increase in the Bjøntegaard-Delta bitrate (BDBR), the drop in the Bjøntegaard-Delta PSNR (BDPSNR), and the amounts of time saved (denoted by TS) are shown when the FME skip algorithm is applied to the HM8.2 reference software. The definitions of BDBR and BDPSNR were presented in earlier work [24]. The average time saved is 12.88%, whereas the increase in the BDBR and the drop in the BDPSNR are 0.33% and 0.011dB, respectively. The least time saved is 8.05% for PartyScene, and the most time saved is 16.47% for Mobisode2. However, the difference in the time savings among all videos is not large. From the sixth to the eighth columns, the R-D performance and the time savings when the part-skip algorithm is applied are shown. The average time saving is 6.37%, whereas the increase in the BDBR and the drop in the BDPSNR are 0.32% and 0.010dB, respectively. Like the simulation results of the FME skip algorithm, the part-skip algorithm also achieves a similar amount of time saved in most of the videos.

In Fig. 6, the rate of FME skip according to the depths and PU sizes is analyzed. The vertical axis represents the rate of FME skip, whereas the horizontal axis represents the PU sizes. D0, D1, D2, and D3 denote depths of 0, 1, 2, and 3, respectively. Figs. 6(a) and (b) are the best two in terms of time saved, whereas Figs. 6(d) and (e) are the worst two for time saved. In all cases, the FME skip probability is much higher in the 2N×2N PU than in the N×2N or the 2N×N PU. The FME skip at a depth of three occurs less frequently than it does in the other depths, but the difference among the depths is not large.

In Fig. 7, the part-skip rate according to depths and PU sizes is shown. The vertical axis represents the part-skip rate, whereas the horizontal axis represents the depth. Between the N×2N and 2N×N PUs, the difference in the part-skip rate is marginal. Similar skip rates are shown for depths 0, 1, and 2; however, part skips rarely occur at a depth of three. The SATD cost at a depth of three is very small. Thus, it is difficult to apply the part-skip condition in Eq. (2) and find the PUs that are expected to have a higher SATD cost than the best SATD cost.

From the third to the fifth columns in Table 2, the R-D performance and the time savings are shown when all of the proposed algorithms of FME skip and part skip are applied to the HM8.2 reference software. The average time saving is 16.47%, whereas the increase in the BDBR and the drop in the BDPSNR are 0.89% and 0.028dB, respectively. From the sixth to the eighth columns, simulation results are shown when the HM fast algorithms of ESD, ECU, and CFM are used to speed up the encoding time. The encoding time is reduced by 16.98%, whereas the increase in the BDBR and the drop in the BDPSNR are 0.82% and 0.025dB, respectively. As shown in Table 2, the performance of the proposed algorithms is comparable to that of HM fast algorithms. From the ninth to the eleventh columns, the R-D performance and the time savings are shown when the HM fast and the proposed algorithms are used together. The average time saving is 36.13%. Compared to when the proposed algorithms and HM fast algorithms are applied separately, the encoding

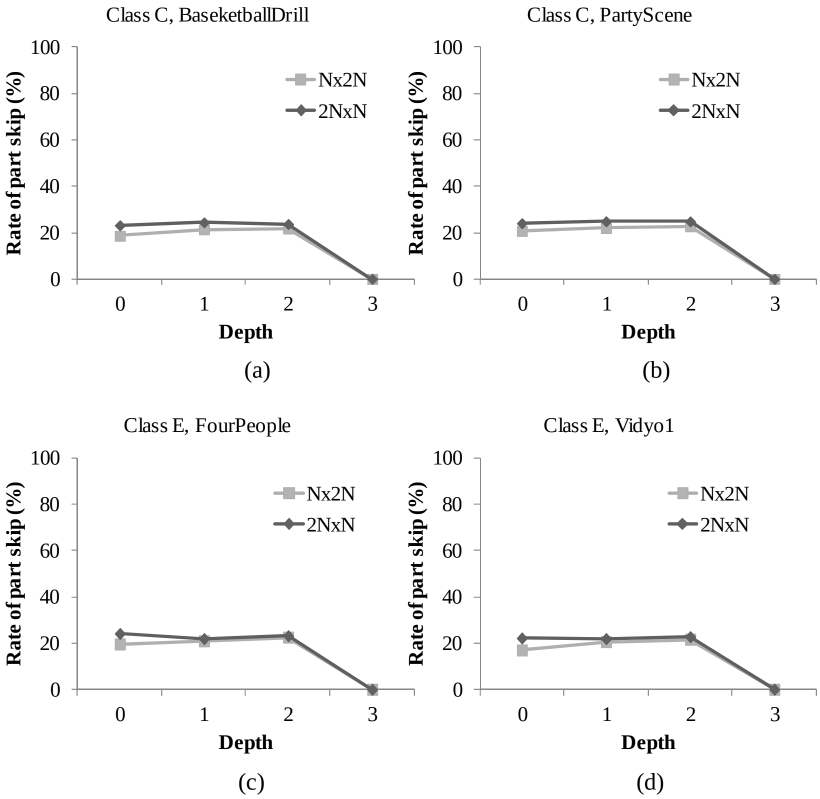

Fig. 7. The rate of PU in which one part is skipped (a) BasketballDrill in Class C, (b) PartyScene in Class C, (c) FourPeople in Class E, (d) Vidyo1 in Class E.

time is significantly reduced, whereas the additional BDBR increase and BDPSNR drop are just 0.37% and 0.011dB, respectively.

5. Conclusion

The main contribution of the proposed algorithms is that the fast decision points do not overlap those proposed in previous studies. Thus, the proposed algorithms are easily used with other fast-decision algorithms, and the consequent speed-up effect is quite independent. In particular, operation-based exploration for early termination of the PU search is very effective for speed-up, with a marginal loss of compression efficiency. In this paper, operation-based early termination is applied only to inter-predictions. However, it would also benefit intra-predictions, as intra-predictions in HEVC become more complicated compared to the previous standard.

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-2015R1C1A1A0 2037625)

References

[1] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264-ISO/IEC 14496-10 AVC), 2003. Article (CrossRef Link)

[2] ISO/IEC JTC 1 SC29 WG11, "Joint Call for Proposals on Video Compression Technology," Doc. N11113, Jan. 2010. Article (CrossRef Link)

[3] ISO/IEC JTC 1 SC29 WG11, "Vision, Applications and Requirements of High-Performance Video Coding," Doc. N11096, Jan. 2010. Article (CrossRef Link)

[4] T. Wiegand, W.J. Han, B. Bross, J. R Ohm, and G.J. Sullivan, "WD4: Working Draft 4 of High-Efficiency Video Coding," JCTVCF803, Torino, IT, July 2011. Article (CrossRef Link)

[5] X. Lu, A.M. Tourapis, P. Yin, and J. Boyce, "Fast Mode Decision and Motion Estimation for H.264 with a Focus on MPEG-2/H.264 Transcoding," *in Proc. International Symposium on Circuits and Systems*, vol. 2, pp.1246-1249, May 2005. Article (CrossRef Link)

[6] C.E.Rhee, J.-S. Kim, and H.-J. Lee, "Cascaded Direction Filtering for Fast Multidirectional Inter-Prediction in H.264/AVC Main and High Profile Compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 403-413, March 2012. Article (CrossRef Link)

[7] B.-G. Kim, and C.-S. Cho, "A fast inter-mode

decision algorithm based on macro-Block tracking for P slices in the H.264/AVC video standard," *in Proc. International Conference Image Processing*, vol. 5, pp. 301-304, Sept. 2007. Article (CrossRef Link)

[8] T. Zhao, H. Wang, and S. Kwong, and C. -C. J. Kuo, "Fast Mode Decision Based on Mode Adaptation," IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 5, pp. 697-705, May 2010. Article (CrossRef Link)

[9] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja, and C. C. Ko, "Fast Intermode Decision in H.264/AVC Video Coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 953-958, July 2005. Article (CrossRef Link)

[10] S.-H. Ri, Y. Vatis, and J. Ostermann, "Fast Inter-Mode Decision in an H.264/AVC Encoder Using Mode and Lagrangian Cost Correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 302-306, Feb. 2009. Article (CrossRef Link)

[11] A. Ahmad, N. Khan, S. Masud, and M.A. Maud, "Selection of variable block sizes in H.264," *in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 173-176, May 2004. Article (CrossRef Link)

[12] H. Zeng, C. Cai, and K.-K. Ma, "Fast Mode Decision for H.264/AVC Based on Macroblock Motion Activity," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 4, pp. 491-499, April 2009. Article (CrossRef Link)

[13] J. Bu, S. Lou, C. Chen, and J. Zhu, "A predictive block-size mode selection for inter frame in H.264," *in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 917-920, May 2006. Article (CrossRef Link)

[14] J. Y. Lee, and H. Park, "A Fast Mode Decision Method Based on Motion Cost and Intra Prediction Cost for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 3, pp. 393-402, March 2012. Article (CrossRef Link)

[15] D. Wu, S. Wu, K. P. Lim, F. Pan, Z. G. Li, and X. Lin, "Block intermode decision for fast encoding of H.264," *in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 181-184, May 2004. Article (CrossRef Link)

[16] Z. Liu, L. Shen, and Z. Zhang, "An Efficient Intermode Decision Algorithm Based on Motion Homogeneity for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 1, pp. 128-132, Jan. 2009. Article (CrossRef Link)

[17] D. Zhu, Q. Dai, and R. Ding, "Fast inter-prediction mode decision for H.264," *in Proc. International Conference Multimedia Expo*, vol. 2, pp. 1123-1126, June 2004. Article (CrossRef Link)

[18] C.-H. Kuo, M. Shen, and C.-C. J. Kuo, "Fast inter-prediction mode decision and motion search for H.264," *in Proc. IEEE International Conference Multimedia Expo*, vol. 1, pp. 663-666, June 2004. Article (CrossRef Link)

[19] P. Yin, H.-Y.C. Tourapis, A.M. Tourapis, and J.Boyce, "Fast mode decision and motion estimation for JVT/H.264," *in Proc. IEEE International Conference on Image Processing*, vol. 3, pp.853-856, Sept. 2003. Article (CrossRef Link)

[20] A. C. W. Yu, G. R. Martin, and H. Park, "Fast Inter-Mode Selection in the H.264/AVC Standard Using a Hierarchical Decision Process," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 2, pp. 186-195, April 2009. Article (CrossRef Link)

[21] T.-C. Chen, Y.-W. Huang, and L.-G. Chen, "Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC," *in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. 9-12, May 2004. Article (CrossRef Link)

[22] M. Shao, Z. Liu, S. Goto, and T. Ikenaga, "Lossless VLSI oriented full computation reusing algorithm for H.264/AVC fractional motion estimation," *IEIEC Trans. Fundamentals*, vol.90-A, no.5, pp. 756-763, April 2007. Article (CrossRef Link)

[23] Y. Song, M. Shao, Z. Liu, S. Li, L. Li, T. Ikenaga, and S. Goto, "H. 264/AVC fractional motion estimation engine with computation reusing in HDTV1080p real-time encoding applications," *in Proc. IEEE Workshop on Signal Processing Systems*, pp.509-514, Oct. 2007. Article (CrossRef Link)

[24] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," presented at the 13th VCEG-M33 Meeting, Austin, TX, Apr. 2001. Article (CrossRef Link)

Chae Eun Rhee received the B.S., M.S. and Ph.D. degrees in Electrical Engineering and Computer Science from Seoul National University, Seoul, Korea, in 2000, 2002 and 2011, respectively. From 2002 to 2005, she was with the Digital TV Development Group, Samsung Electronics Company Ltd., Suwon City, Korea, as an Engineer, where she was involved in bus architecture and MPEG decoder development. In 2013, she joined the Department of Information and Communication Engineering at Inha University, Korea, where she is currently working as an assistant professor. Her research interests include algorithm and architecture design of video coding for HEVC and H.264/AVC and configurable video coding for real time systems.