International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 10, No. 7 (2017), pp. 105-118
http//dx.doi.org/10.14257/ijsip.2017.10.7.10

A Fast Inter Prediction Method in HEVC Using SAD Computing Algorithm Based on Bottom-up Design

Sunghun Jeon, Seungyong Park and Kwangki Ryoo

*Adept. of Information & Communication Eng, Hanbat National University
125 Dongseodaero, Yuseong-gu, Dajeon 305-719, Republic of Korea
Jsh4189@naver.com, srrr.kr@gmail.com, kkryoo@gmail.com*

Abstract

In this paper, we propose an efficient algorithm for computing architecture for high-performance Inter Prediction SAD encoder. HEVC Motion Estimation(ME) Inter prediction is a process of searching for the currently high prediction block (PU) and the correlation in the interpolated reference picture in order to remove temporal redundancy. Conventional ME algorithm uses full search(FS) or fast search algorithm. Full search technique has the guaranteed optimal result but has many disadvantage which include high calculation and increase in operational time due to the motion prediction with respect to all candidate blocks in a given search area. Therefore, this paper proposes a new algorithm which reduces the computational complexity by reusing the SAD operation in full search to reduce the amount of calculation and computational time of Inter Prediction. The proposed algorithm is applied to an HEVC standard software HM16.12. There was an improved operational time of 61% compared to the traditional full search algorithm, BDBitrate was decreased by 11.81% and BDPSNR increased by about 0.5%.

Keywords*: HEVC, Inter Prediction, SAD*

1. Introduction

With the development of various electronic devices, it is possible to acquire 4K UHD (Ultra High Definition) image. In addition, with the release of UHDTV (Ultra High Definition TV) following HDTV, ultra-realistic and high-quality broadcast service is provided, and as smart-phones with HD resolution as its basic screen resolution is launched in the mobile market, the user demand for the high quality products is increasing rapidly [1]. In this context, the International Telecommunication Union (ITU-T) and the International Organization for Standardization / International Electrotechnical Commission (ISO / IEC) jointly developed High Efficiency Video Coding (HEVC), the next generation video coding standard. HEVC compresses high-resolution images at a higher efficiency than that of H.264 / AVC which is the standard video compression technology used [2-4].

In the HEVC, CU (Coding Unit), PU (Prediction Unit) and TU (Transform Unit) are the three kinds of coding units are applied and a coding of a hierarchical quad-tree structure is performed. Also the coding units of various sizes ranging from 64×64 to 8×8 are used. In addition, by applying intra-picture prediction direction, enhanced motion estimation, motion vector merging, and fine-grained loop filter, Compression performance of ship is shown, but computational complexity greatly increases with various coding structures and advanced prediction techniques [5].

New techniques of motion prediction methods take into consideration the current characteristics of the original pixels and sub-pixels in the process of comparing the correlation of the PU and the reference block in order to generate the current frame and the most similar prediction frame measurement using SAD (Sum of Absolute Difference). However, in the case of Full Search for inter prediction, there is high amount of

ISSN: 2005-4254 IJSIP

calculations and computational time is increased because the SAD operation is performed on all block sizes of the PU(original) from 64x64 down to 4x4 and a repeat of the SAD calculation is also done for the reference search area [6].

In this paper, we propose a new algorithm that reduces the amount of calculation and computational time of the SAD operation for high-performance HEVC Inter Prediction. The proposed algorithm stores the SAD value of the result of the previous SAD operation iteration to a minimum block size of 4x4 units. One 4x4 block SAD results are stored and reused for every block size. SAD operation division is performed after inter prediction motion estimation [7].

This paper is organized as follows: Section 2 describes Inter-prediction technology standard of HEVC and Section 3 describes the proposed inter-prediction algorithm. A performance comparison of the proposed algorithm is described in Section 4 and finally Section 5 concludes the study.

2. Overview of HEVC Encoding Process

2.1. Coding Structure

HEVC has adopted three kinds of tree-structured unit representations: a CU, prediction unit (PU), and transform unit (TU). CU is the basic unit of region splitting for intra and inter predictions, PU is the basic unit of the prediction processes, and TU is the basic unit of the transform and quantization processes [8].

The input image is encoded on a block-by-block basis, and the original block is predicted from the prediction block generated through intra-picture prediction or inter-picture prediction. The difference between the original block and the prediction block is referred to as a difference block. The difference block sequentially undergoes transformation, quantization, and entropy encoding.

2.2. Motion Estimation

Each picture constituting a video has a high correlation with each other in terms of time. Accordingly, a prediction value for a coding block in a current picture to be coded can be generated from a picture that has already been coded at a previous time. The technique of generating the prediction block from the picture coded at the previous time is referred to as inter prediction.

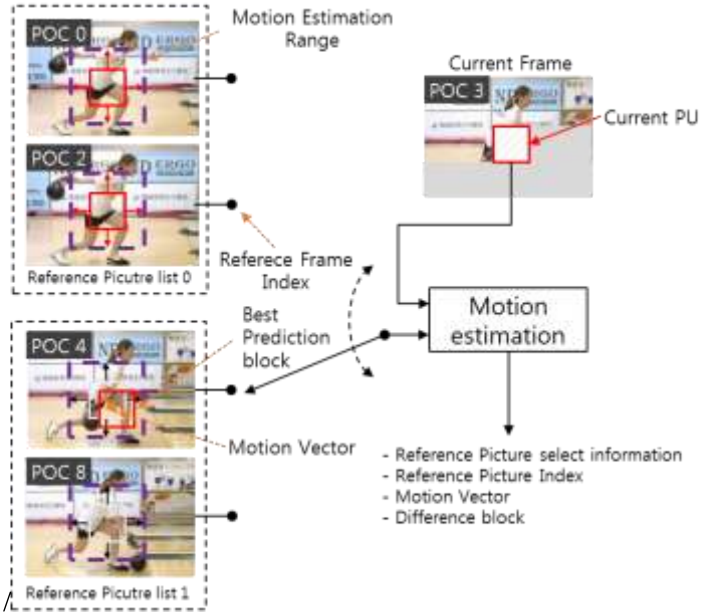

Figure 1. HEVC Motion Estimation

Figure 1 shows an example of the motion estimation process. The motion estimation for performing the inter-picture prediction is a process performed in the encoder, and is a process for searching a predicted block having a high degree of correlation with the current PU in the interpolated reference picture [9]. As a result of the motion estimation, coefficients obtained by transforming and quantizing the information of the reference picture list, the reference picture index, the motion vector, and the difference signal in units of PU are transmitted to the decoder, and the decoder uses the surrounding information transmitted from the encoder, and generates a reconstructed block using the quantized residual signal [10].

When measuring the correlation between the reference block and the current block, the encoder uses a simple correlation measurement method in consideration of complexity. Formula (2.1) represents the Sum of Absolute Difference (SAD) used in the correlation measurement of integer pixels and Formula (2.2) represents the Sum of Transform Difference (SATD) used in subpixel correlation measurement.

$$SAD(i,j,k,l) = \Sigma \,|\, (B_{cur}(i,j) - B_{ref}(k,l)) \,| \qquad (1)$$

$$SATD(i,j,k,l) = \Sigma \,|\, (T(B_{cur}(i,j) - B_{ref}(k,l))) \,| \qquad (2)$$

B_{cur} denotes the current block, B_{ref} denotes the motion estimation candidate block existing in the reference picture, i, j denotes the position of the current PU, and l denotes the PU position of the motion estimation target [11].

One PU block has eight division patterns as shown in Figure 2 and selects the best one. Part_NxN can split the CU block to the minimum size (8x8). However, AMP (Asymmetric Motion Partition) such as PART_2NxnU, Part_2NxnD, Part_nLx2N and Part_nRx2N cannot split CU block to the minimum size because of the complexity of encoding and decoding [12-14].

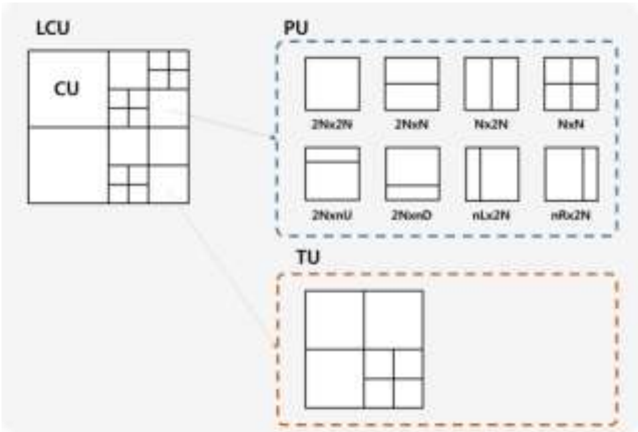

Figure 2. PU Split Screen Mode of Inter Prediction

2.3. Search Algorithms

2.3.1. Full-Search Algorithm

The full-search method of motion estimation is a method of comparing the reference block with all the blocks within the predetermined search range. Figure 3 shows the full-search range in a 64×64 PU block. Compared to all blocks in the search range as shown in the figure, the computational time is very high and takes more than 96% of the actual encoding time [15]. Therefore, many search algorithms have been proposed to reduce computation time, and HEVC standard software HM adopts TestZoneSearch (TZS) algorithm and full-search algorithms as its standard.

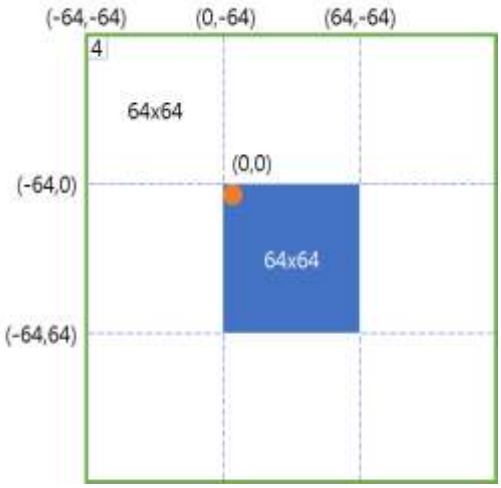

Figure 3. 64×64 PU Block Full-search Search Range

2.3.2. TZS Algorithm

The TZS algorithm calculates only 4 or 8 search points in a grid search. The search distance of each pattern is increased by 2 times in each steps, increasing the search distance from 1 to 8 and searching from 1 to 64 of the search area. After the search, the point with the smallest SAD value is set as the midpoint of the next search step, and the search distance is stored. Figure 4 shows the 8-point diamond search in the TZS algorithm.

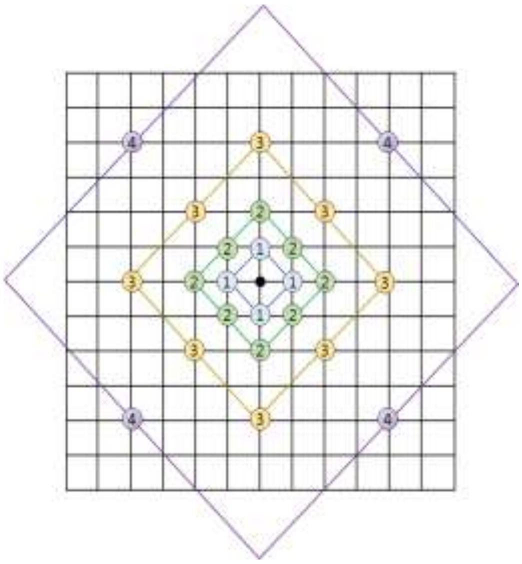

Figure 4. 8-point Diamonds Search

The TZS algorithm finds the motion vector position by calculating all the points in the search range as a full-search. However, since only 4 or 8 search points are calculated, the computational time is reduced but the prediction performance is lower than that of Full-search.

3. Proposed Motion Estimation Algorithm

Table 1 shows the encoding time for the main functions of the HM encoder.

Table 1. HM Function Encoding Time

Function	Time(%)	
	All Intra	Random Access
TEncSearch	11.8	7.4
TComTrQuant	24.4	10.7
TComRdCost	9.8	38.8
TComInterpolationFilter	0.0	19.8
TComYUV	0.1	1.7
partialButterfly	8.7	4.0
TComDataCU	5.8	2.7
TEncSbac	8.4	3.5
TEncEntropy	1.2	0.6
TEncBinCABAC	2.2	0.9
TCOMPrediction	10.0	1.1
TComPattern	6.6	0.4
Memcpy/memset	11.0	7.1

In the case of All Intra mode, which only performs intra prediction, the TComRdCost which is the function that performs the SAD operation takes only 9.8% of the entire encoding speed. However, in the case of Random Access which performs the inter prediction, it takes 38.8% of the entire encoding speed. This is because the number of repetitions of the SAD operation used for finding an optimal block in inter-picture prediction motion estimation is large.

In the proposed SAD algorithm, the maximum CU block size is 64x64. A 4x4 block size is utilized to store the SAD operation results for re-use in all SAD operation to determine the optimal size for PU mode.

Table 2. PU Block Sizes in Inter Prediction

PU SIZE	Depth			
	0	1	2	3
2N x 2N	64x64	32x32	16x16	8x8
2N x N	64x32	32x16	16x8	8x4
N x 2N	32x64	16x32	8x16	4x8
N x N	32x32	16x16	8x8	
2N x nU	64x16	32x8	16x4	
2N x nD	64x48	32x16	16x12	
nL x 2N	16x64	8x32	4x16	
nR x 2N	48x64	16x32	12x16	

The original full-search SAD algorithm computes all possible values for the search range. Therefore, the SAD operation is performed repeatedly causing redundancy and increasing the complexity which consistently lowers the overall encoding time. Table 2 shows the PU block for maximum CU block size of 64x64 and Figure 5 shows the original full-search algorithm flow.

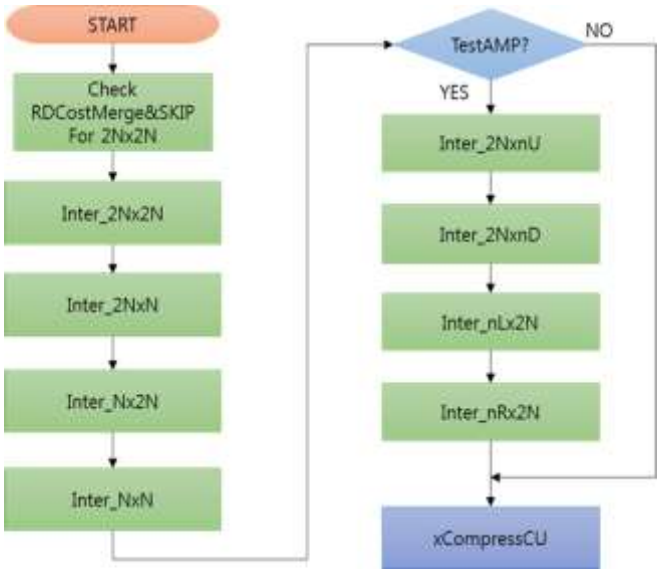

Figure 5. Inter Mode Decision Flow

In order to reduce the repeatability of the SAD operation process, the proposed SAD algorithm stores the SAD value in a 4x4 block size for reuse. The sum of the SAD result of the four 4x4 blocks as shown in Figure 6 is the same as the SAD value of the 8x8 block and thus the result of the four 8x8 blocks create the same output as the SAD value of the 16x16 block. In this regard, all 23 categories as shown in the PU table for the MAX CU of 32x32 block size can be achieved via the 4x4 unit SAD operation.

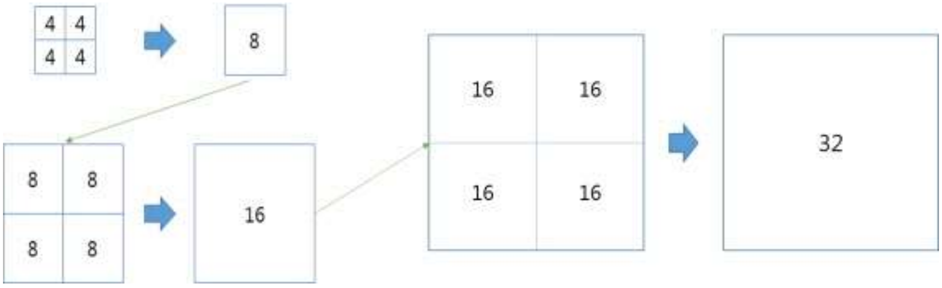

Figure 6. Method of Block Re-use

Figure 7 shows the overall flow of the proposed inter-picture prediction for block reuse. XCheckRDCostMerge2N×2N calculates the RD-Cost for skip and Merge. When 2N×2N SAD operation is performed instead of calculating RD-Cost in order of PU split mode such as 2N×2N, N×N, N×2N, 2N×N, *etc.*, SAD result is stored in 4×4 unit in memory. Next, in order of N×N, N×2N, and 2N×N PU mode, memory indexes of the corresponding area are called in order to obtain the result value. In the process of calculating the RD cost for the last AMP (Asymmetric Motion Partition) partition, the index of the corresponding area is called to determine the optimal PU mode.

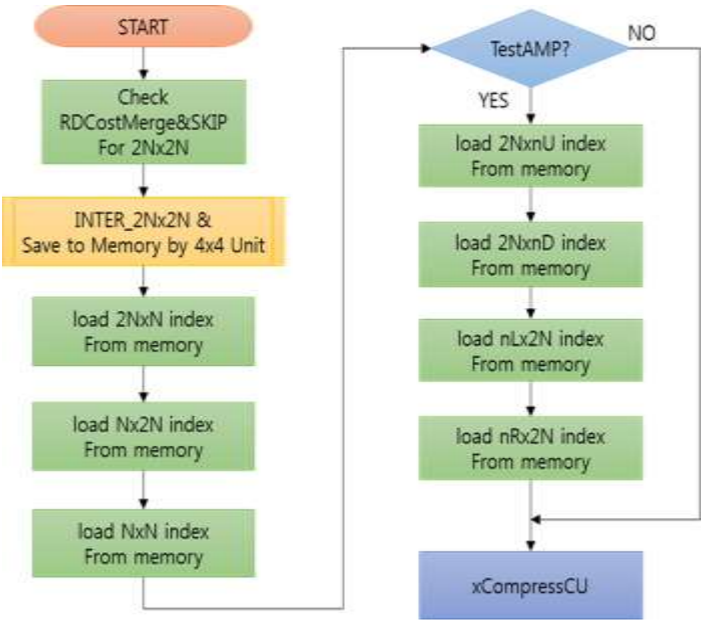

Figure 7. Proposed Algorithm Flow Chart

When the MAX CU size is 32×32, 64 SAD result values are stored in memory in 4×4 units as shown in Figure 8. For the two 32×16 SAD operations, 2N×N to be executed thereafter, the SAD operation results obtained from (1 to 32) and (33 to 64) are searched and calculated. For all subsequent block segmentation, the 4 0x4 unit result value of the corresponding area is called and reused to lower the overlapping rate of the SAD operation in the same area.

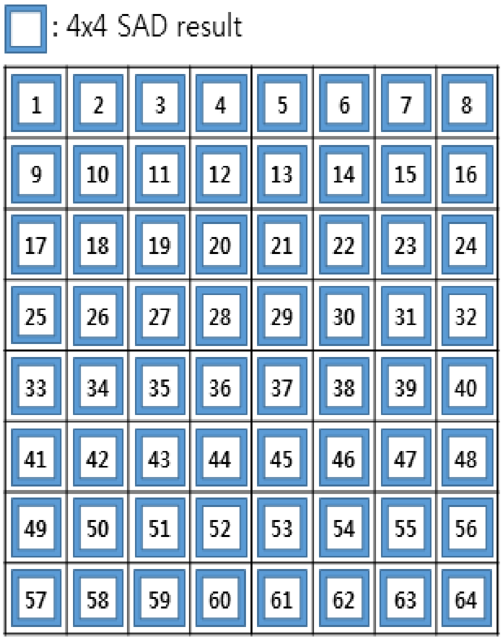

Figure 8. 32×32 PU Index

3. Experiment Results

The proposed method has been implemented based on the HM16.12 encoder which is used as an anchor in the experiments. Our experiments are performed on the lowdelay_P_main10 encoding configuration. We compare the performance improvement of the proposed algorithm at various resolutions. The conditions used in the experiments are shown Table 3.

Table 3. System Environment

Classification	specification
CPU	Intel Core i7-3770 CPU@ 3.40 GHz
RAM	Samsung 8.00GB Memory
HDD	Samsung SSD 840
GPU	-

Figure 9 shows the motion vector analysis of the bin file generated by applying the HM16.12 standard software and the proposed algorithm to the BasketballPass.yuv image using the Elecard HEVC Analyzer tool.

(a) HM.16.12 BasketballPass

(b) Proposed BasketballPass

Figure 9. Motion Vector Analysis Using Analyzer

Although motion estimation similar to the HM standard is performed in most block regions, fine motion estimation is performed in a portion where a lot of motion occurs and is divided into small PU blocks.

Table 4. CLASS A Verification Result

sequence	QP	HM16.12			Proposed			BD rate	BD PSNR	∆TS (%)
		Bitrate (kbps)	PSNR (dB)	Time	Bitrate (kbps)	PSNR (dB)	Time			
Traffic	22	36923.9	41.9	35944.5	34627.0	41.9	14374.8	0.49	-12.26	-60.0
	27	17512.3	39.2	33872.3	15992.5	39.3	14171.1			-58.2
	32	9200.7	36.7	32775.7	8298.4	36.9	13887.1			-57.6
	37	5010.9	34.1	30163.4	4471.9	34.4	12358.4			-59.0
People on street	22	57089.2	41.8	39346.6	56521.0	41.8	14887.3	0.18	-3.76	-62.2
	27	29171.9	38.7	37234.9	28554.6	38.7	14441.6			-61.2
	32	15537.3	35.5	34819.8	15078.4	35.6	14147.3			-59.4
	37	8775.2	32.7	30712.1	8404.2	32.9	13928.6			-54.6
Nebuta	22	320784.3	39.4	36613.4	309189.0	39.4	16633.1	0.34	-6.96	-54.6
	27	184040.4	35.8	38405.5	176461.7	35.9	15811.5			-58.8
	32	82086.5	31.3	37277.2	76515.6	31.4	14920.9			-60.0
	37	31787.7	28.3	35732.2	29136.2	28.6	14403.7			-59.7
Steam Locomotive	22	13300.5	40.7	39678.5	128369.7	40.7	14473.2	0.15	-4.61	-63.5
	27	50405.3	37.2	36712.0	48041.9	37.2	13293.6			-63.8
	32	23395.9	35.1	34894.9	23205.2	35.2	12908.3			-63.0
	37	11851.5	32.9	30830.8	11799.6	33.0	12715.2			-58.8

Table 5. CLASS B Verification Result

sequence	QP	HM16.12			Proposed			BD rate	BD PSNR	ΔTS (%)
		Bitrate (kbps)	PSNR (dB)	Time	Bitrate (kbps)	PSNR (dB)	Time			
Kimono	22	12724.5	42.7	18832.2	9987.0	42.7	7503.3	1.36	-30.02	-60.2
	27	6893.9	40.7	18247.3	5126.2	40.8	7105.5			-61.1
	32	3769.0	38.1	17538.6	2739.0	38.5	6956.9			-60.3
	37	2033.7	35.4	17271.2	1488.7	36.1	6931.7			-59.9
Park scene	22	18352.1	40.8	21026.0	16995.1	40.8	7545.4	0.40	-11.34	-64.1
	27	8612.7	38.2	19874.2	7950.9	38.2	7182.3			-63.9
	32	4163.6	35.6	18610.0	3834.2	35.8	7003.8			-62.4
	37	1968.0	33.3	17264.1	1801.9	33.5	7068.3			-59.1
Cactus	22	46174.8	39.4	20764.8	43968.0	39.4	7556.9	0.31	-11.21	-63.6
	27	17443.5	37.4	19290.2	16042.6	37.4	7132.8			-63.0
	32	8656.6	35.3	18549.1	7984.7	35.5	7009.9			-62.2
	37	4474.2	33.2	17358.4	4104.8	33.4	6877.6			-60.4
BQ terace	22	92743.3	40.3	23152.4	92316.9	40.3	7783.2	0.02	-0.58	-66.4
	27	31746.0	36.5	20865.7	31673.2	36.6	7163.6			-65.7
	32	13909.1	34.4	1937.1	13862.2	34.4	6998.4			-63.9
	37	6751.8	32.2	16850.6	6757.6	32.2	6893.1			-59.1

Table 6. CLASS C Verification Result

sequence	QP	HM16.12			Proposed			BD rate	BD PSNR	ΔTS (%)
		Bitrate (kbps)	PSNR (dB)	Time	Bitrate (kbps)	PSNR (dB)	Time			
Race horses	22	11571.6	40.5	3925.4	10048.7	40.4	1707.5	1.16	-21.18	-56.5
	27	5849.9	36.2	3744.1	4867.7	36.4	1639.8			-56.2
	32	2910.5	32.7	3579.3	2396.7	33.2	1590.5			-55.6
	37	1479.9	30.0	3250.7	1145.3	30.5	1514.4			-53.4
BQ mall	22	9632.9	40.8	3303.1	9321.9	40.8	1587.7	0.23	-4.66	-51.9
	27	5171.1	38.1	3197.4	5002.6	38.1	1543.1			-51.7
	32	2898.5	35.2	3138.7	2804.4	35.3	1516.8			-51.7
	37	1564.8	32.2	3068.8	1512.9	32.3	1503.6			-51.0
Party scene	22	16907.3	40.3	3949.5	16898.6	40.3	1668.9	0.02	-0.26	-57.5
	27	9617.9	36.3	3858.2	9598.1	36.3	1605.0			-55.7
	32	5301.7	32.8	3721.7	5300.6	32.8	1571.9			-53.9
	37	2786.3	29.6	3412.8	2774.2	29.6	1530.7			-51.3
Basketball drill	22	6632.6	41.8	3829.3	6435.8	41.8	1627.9	0.27	-5.98	-57.5
	27	3435.3	38.6	3568.4	3294.4	38.6	1582.1			-55.7
	32	1792.1	35.8	3372.5	1661.4	35.8	1555.8			-53.9
	37	978.2	33.3	3146.5	883.3	33.3	1532.9			-51.3

Table 7. CLASS D Verification Result

sequence	QP	HM16.12 Bitrate (kbps)	HM16.12 PSNR (dB)	HM16.12 Time	Proposed Bitrate (kbps)	Proposed PSNR (dB)	Proposed Time	BD rate	BD PSNR	∆TS (%)
Race horses	22	3099.2	40.2	1164.6	2730.1	40.2	350.2	1.17	-18.70	-69.9
	27	1720.3	35.9	1165.2	1482.0	36.2	331.2			-71.6
	32	905.9	32.1	935.0	765.5	32.5	318.5			-65.9
	37	456.1	29.0	1162.9	383.1	29.7	310.2			-73.3
BQ square	22	4727.4	40.2	1188.1	4737.2	40.2	339.6	0.00	0.04	-71.4
	27	2446.3	36.1	1134.5	2453.7	36.1	323.1			-71.5
	32	1344.8	33.1	875.9	1343.9	33.1	310.9			-64.5
	37	758.3	30.3	1121.3	760.2	30.3	305.6			-72.7
Blowing bubbles	22	3554.6	39.5	1182.3	3581.8	39.6	332.2	0.01	-0.21	-71.9
	27	1783.5	36.1	1166.2	1806.3	36.2	318.4			-72.7
	32	899.4	33.1	925.7	905.7	33.2	312.4			-66.3
	37	453.7	30.4	1161.5	446.9	30.4	302.4			-74.0
Basketball pass	22	1124.6	42.2	796.6	1127.7	42.2	312.5	0.06	-1.14	-60.8
	27	612.1	38.6	776.4	607.3	38.6	306.3			-60.6
	32	325.7	35.3	751.4	320.1	35.3	302.4			-59.8
	37	171.6	32.5	725.5	165.8	32.6	297.6			-59.0

Table 4 and7 shows the experimental results of the proposed algorithm. The performance of our method in terms of the change in bitrate, peak signal-to-noise ratio (PSNR), and total encoding time are reported based on the following formulae:

$$\Delta Bitrate = Bitrate(proposed) - Bitrate(anchor) \tag{3}$$

$$\Delta PSNR = Y_{PSNR}(proposed) - Y_{PSNR}(anchor) \tag{4}$$

$$\Delta Time(\%) = \frac{Time(\Pr oposed) - Time(anchor)}{Time(anchor)} \tag{5}$$

There was an average decrease of 61% in encoding time compared to standard HM.16.12 using equation (5). In addition, BD-PSNR increased by 0.515 and BD-Bitrate reduced by 11.81.

Table 8. Comparison with Existing Research

Sizes	Sequences	proposed BDrate	proposed ∆TS(%)	[16] BDrate	[16] ∆TS(%)	[17] BDrate	[17] ∆TS(%)
ClassA	Traffic	-12.26	60.5	1.88	60.2	1.67	45.3
	PeopleOnstreet	-3.76	59.3	1.11	41.2	1.44	29.2
ClassB	Kimono	-30.02	60.4	1.93	53.2	1.04	36.7
	ParkScene	-11.34	62.3	2.09	55.2	0.86	44.4
	Cactus	-11.21	62.3	1.65	55.5	1.20	39.7
	BasketballDrive	-44.55	62.2	2.22	56.6	1.27	40.4
ClassC	RaceHorsesC	-21.18	55.4	1.29	40.9	1.13	26.8
	BQMall	-4.66	52.8	1.70	52.6	1.27	42.2
	PartyScene	-0.26	57.3	2.47	40.3	0.99	31.2

	BasketballDrill	-5.98	54.6	0.96	48.3	1.58	37.2
ClassD	RaceHorses	-18.70	70.2	1.46	30.3	0.97	23.0
	BlowingBubbles	-0.21	71.2	2.82	42.6	1.17	32.7
	BasketballPass	-1.14	60.0	3.30	53.1	0.84	40.7
	BQSquare	0.00	70.0	0.75	52.0	1.30	32.6
Average	-	-11.81	61.32	1.83	48.7	1.20	35.9

Table 8 shows the comparison between the motion estimation algorithm proposed in the previous papers and the proposed algorithm. Compared with the [19] and [20], the encoding speed improved by about 19% and BDBitrate decreased by 13.32.

4. Conclusion

In this paper, we propose a new algorithm based on global search in order to minimize computation time and computation complexity of SAD computation in HEVC motion estimation algorithm. Since the SAD operation used in the conventional motion estimation algorithm is performed for all block division ranging from (64×64) to (4×4), there are many repetitive operations. Therefore, in this paper, we apply the bottom-up algorithm that saves the SAD results in 4×4 units and re-uses the stored results in the upper blocks. Compared with the global search method applied to the HM standard software HM-16.12, the proposed algorithm improved on average 61% encoding speed, BDPSNR increased by 0.5, and BDBitrate decreased by 11.81.

Acknowledgment

This research was supported by the MSI(Ministry of Science, ICT and Future Planning), Korea, under the Global IT Talent support program(IITP-2016-R0134-16-1019) and Human Resource Development Project for Brain scouting program(IITP-2016-R2418-16-0007) supervised by the IITP(Institute for Information and Communication Technology Promotion)

References

[1] ITU-T Q6/16 and ISO/IEC JTC1/SC29/WG11, "Joint Call for Proposals on Video Compression Technology", ITU-T SG16/Q6/VCEG-AM91, 39th VCEG, Kyoto, Japan, **(2010)**.

[2] ITU-T SG16 WP3 and ISO/IEC JCT1/SC29/WG11, "High Efficiency Video Coding(HEVC) text specification draft 10 (for FDIS & Content)", JCTVC document, L1003, Geneva, CH, **(2013)**.

[3] I. E. G. Richardson, "H.264 and MPEG-4 Video Compression", Wiley, **(2003)**.

[4] G. J. Sullivan, J. R. Ohm, and W. Han, "Overview of the high efficiency video coding(HEVC) standard", IEEE Trans. on Circuits Syst. Video Technol., vol. 22, no. 12, **(2012)**, pp. 1649-1668.

[5] V. Sze, M. Budagavi and G. J. Sullivan, "High Efficiency Video Coding(HEVC) Algorithms and Architectures", Springer, **(2015)**.

[6] C. C. Lien and C. P. Yu, "A Fast Mode Decision Method for H.264/AVC Using the Spatial Temporal Prediction Scheme", Proc. 18th IEEE Int. Conf. Pattern Recog., vol. 4, **(2006)**, pp. 334-337.

[7] J. Xin, A. Vetro, and H. Sun, "Efficient Macroblock Coding-Mode Decision for H.264/AVC Video Coding", Proc. 24th Picture Coding Symp., **(2004)**.

[8] I. E. G. Richardson, "Video codec design", Wiley, **(2003)**, pp. 93-124

[9] B. Shen and I. K. Sethi, "Direct Feature Extraction from Compressed Images", Proc. IS&T/SPIE Conf. Storage Retrieval Image Video Databases, vol. 2670, **(1996)**, pp. 404-414.

[10] JCT-VC, "High Efficiency Video Coding (HEVC) Test Model 2 (HM 2) Encoder Description", Document JCTVC-D502, **(2011)**.

[11] F. Sampaio, S. Bampi, M. Grellert, L. Agostini, and J. Mattos, "Motion vectors merging: low complexity prediction unit decision heuristic for the inter-prediction of HEVC encoders", IEEE International Conference on Multimedia and Expo (ICME), pp. 657-662, **(2012)**.

[12] JCT-VC, "WD2: Working Draft 2 of High-Efficiency Video Coding", Document JCTVC-D503, **(2011)**.

[13] F. Bossen, "Common Test Conditions and Software Reference Configurations", Document JCTVC-F900, Torino, IT, **(2011)**.
[14] D. Jun and H. Park, "An Efficient Priority-Based Reference Frame Selection Method for Fast Motion Estimation in H.264/AVC", IEEE Trans. Circuits Syst. Video Technology, vol. 20, no. 8, **(2010)**, pp. 1156-1161.
[15] N. Purnachand, L. N. Alves, and A. Navarro, "Improvements to TZ search motion estimation algorithm for multiview video coding", 2012 19th International Conference on System, Signals and Image Processing(IWSSIP), **(2012)**, pp. 388-391.
[16] J. Zhang, B. Li and H. Li, "An Efficient Fast Mode Decision Method for Inter Prediction in HEVC", IEEE trans. Circuits, Syst. Video Technology, vol. 26, no.8, **(2016)**, pp. 1502-1515.
[17] S. Ahn, M. Kim, and S. Park, "Fast decision of CU partitioning based on SAO parameter, motion and PU/TU split information for HEVC", Procedure, Picture Coding Symposium, **(2013)**, pp. 113-116.

Authors

Sunghun Jeon, received a BSc Degree in Information and Communication Engineering from Hanbat National University, South Korea, in 2015. He is currently pursuing a MENG Degree in Information and Communication Engineering at Hanbat National University, South Korea. His research interests include SoC Design and Verification Platforms and Communication Architectures.

Seungyong Park, received a BS Degree in Information and Communication Engineering from Hanbat National University, South Korea, in 2010 and MENG Degree in Information and Communication Engineering from Hanbat National University, South Korea in 2012. He is currently pursuing a PhD Degree in Information and Communication Engineering at Hanbat National University, South Korea. His research interests include SoC Design and Verification Platforms, Image Signal Processing and Multimedia Codec Design.

Kwangki Ryoo, received BS, MS and PhD Degrees in Electronic Engineering from Hanyang University, South Korea in 1986, 1988 and 2000 respectively. From 1991 to 1994, he was an Assistant Professor at the Korea Military Academy in South Korea. From 2000 to 2002, he worked as a Senior Researcher at Electronics and Telecommunication Research Institute, South Korea. From 2010 to 2011, he was a Visiting Professor at University of Texas at Dallas. Since 2003, he has been a Professor at Hanbat National University, South Korea. His research interests include Engineering Education, SoC Design and Verification, Image Signal Processing and Multimedia Codec Design.

