Context-aware block-based motion estimation algorithm for multimedia internet of things (IoT) platform

Avishek Saha[1] · **Young-Woon Lee**[1] · **Young-Sup Hwang**[1] · **Kostas E. Psannis**[2] · **Byung-Gyu Kim**[3]

Received: 23 February 2017 / Accepted: 23 June 2017 / Published online: 8 July 2017

Abstract Shaping video data into fast-responding transmission and high resolution output video using cost-effective video processing is desirable in many applications including Internet of Things (IoT) applications. In association with rapid development of IoT smart sensor applications, real-time processing of huge-amount of data for a video signal has become necessary leading to video compression technology. Motion estimation (ME) is necessary for improving the quality, but it has high computational complexity in video compression system. The present article, therefore, proposes a context-aware adaptive pattern-based ME algorithm for multimedia IoT platform to improve video compression. In the proposed algorithm, the motions are classified into large or small based on distortion value. Accordingly, the search pattern is chosen either small diamond search pattern (SDSP) or large diamond search pattern (LDSP) in each and every step of ME; allowing adaptive processing of large and small abstract information. Compared to conventional fast algorithms, the experimental results demonstrate up to 40 and 36% reduction in encoding time for low-delay main (LB-main) and random access main (RA-main) profiles respectively in HEVC test model 16.10 encoder with bit-rate loss of 0.071 and 0.246% for both the profiles, ensuring quality video and searching precision.

Keywords IoT · Block-based motion estimation · Motion degree · Adaptive pattern · HEVC

✉ Byung-Gyu Kim
bg.kim@sm.ac.kr

Avishek Saha
avishek.saha.ofc@gmail.com

Young-Woon Lee
yw.Lee@vicl.sookmyung.ac.kr

Young-Sup Hwang
young@sunmoon.ac.kr

Kostas E. Psannis
kpsannis@uom.edu.gr

[1] Department of Computer Engineering, SunMoon University, Asan, Republic of Korea

[2] Department of Applied Informatics, University of Macedonia, Thessaloniki, Greece

[3] Department of IT Engineering, Sookmyung Women's University, Seoul, Republic of Korea

1 Introduction

IoT is an umbrella term includes the network of electronic devices like smart phones, wearable electronic devices, or connected appliances that are capable of communicating wirelessly with each other [1]. From applicability standpoint, these artificially intelligent networks interact with respective physical environment, allowing IoT to influence the present civilization from home automation to security, health monitoring, and managing the daily task. The growing applicability of IoT is hypothesized to reach over 50 billion devices by 2020 [2]. With the advancement, considerable amount of data traffic is generated from IoT-based devices, which are major multimedia data. The enormous size of raw multimedia data followed by its storage and transmission pushes the urge of its compression using video compression technology. The chief objectives of video compression technique are efficient storing of digital video, effective transfer of visual data among various components of a video system, and reduce computational resources used in video processor for cost-effective computational processing. Video compression involves compression of digital videos by removing the redundant data, which is carried out by ME algorithms. Block-based motion estimation (BME) algorithm has been employed in most recent video coding standards, such as H.264 [3] and H.265/ HEVC [4] due to its simplicity and effectiveness.

While discussing on BME algorithm, first algorithm comes in context is full search (FS) algorithm, the simplest and effective but takes huge time. Jain and Jain [5] proposed a robust and improved BME algorithm, 2D logarithmic search (2DLS). 2DLS takes lesser encoding time than FS, while maintaining the image quality. The main concept is based on reducing the search window (SW) to half in each iteration. Three-Step Search (TSS) Algorithm [6], New Three-Step Search (NTSS) [7], and Four-Step Search Algorithm (FSS/4SS) [8] are the improved version of BME algorithms proposed based on checking the surrounding fix points. Diamond Search (DS) [9] uses fixed diamond pattern to calculate the minimum block distortion (MBD). There are several DS-based algorithms proposing reduced ME complexity by minimizing the search points. Cross-Diamond Search (CDS) [10, 11], Small-Cross-Diamond Search (SCDS) [12] using diamond and square patterns [13], and Small-Diamond-based search algorithm [14] are some of extension of diamond pattern search BME algorithm. Star Diamond Algorithm [15] uses star-shaped pattern incorporating two search patterns; star shape in first step, followed by small diamond. Hexagon-based search algorithm [16] uses big hexagonal pattern at its first step, followed by small hexagon to conclude final motion vector (MV). All these algorithms start their search with fixed pattern, focusing to reduce the number of search points or search direction.

To further reduce the encoding time, a context-aware pattern search BME algorithm is proposed in this paper. The choice of pattern is tossed between large and small based on the degree of the motion that can easily avoid direction misleading, trapping in local minimum, hence, minimize the search time. The article is organized into following chapters. Section 2 provides a details description of the proposed context-aware fast BME algorithm, followed by experimental results and discussion in Section 3. Finally, Section 4 concludes the article with the light of potential future prospect.

2 Proposed method

As discussed in the previous section, all BME algorithms start the search with a fix pattern, focusing either to reduce the number of search points or factoring the motion direction. None of these standard BME algorithms investigate the degree of motion at first point. Very few of them consider the motion degree in the later part of algorithm. Although, these state-of-the-art algorithms are very promising, however, there is high chance of motion misleading and inefficiency to estimate small motion in first step.

To further improve the performance, a context-aware pattern search BME algorithm is proposed here. The main feature of this proposed algorithm is instead of using only a fix pattern, the motions are characterized into large or small based on motion degree at the very first step, and the search pattern is chosen either SDSP or LDSP. This is continued to each and every steps until MBD is at center of SDSP. By this approach, the computation time reduces significantly while maintaining the quality.

Categorization of motion degree is done using distortion ratio (DS_r), which is derived from the pixel differences between current and reference blocks, represent as initial distortion value (DS_p) and final (DS_c) distortion value at initial and final MV points respectively. DS_r is obtained using DS_c and DS_p. The details of the procedure and mechanism of the proposed algorithm are summarized below.

2.1 Determination of the search center

Instead of starting the search from the center of the search window, the proposed algorithm uses advanced motion vector predictor (AMVP) logic to select the initial search center (ISC). Basically, it compares median MV ($\overrightarrow{MV}_{MP}$), $\overrightarrow{MV}_{left}$, $\overrightarrow{MV}_{up}$, $\overrightarrow{MV}_{up-right}$, and $\overrightarrow{MV}_{(0,0)}$ and the point which has MBD, is considered as the final initial search point as described in Fig. 1. The proposed algorithm deploys the search pattern at final ISC and searching process starts.

2.2 Distortion ratio

To classify the motion degree of the video content, distortion value is used in this proposal. Distortion value is derived from the pixel differences between the current and reference blocks. Distortion values gives the motion trends, it helps to define the motion. Distortion ratio (DS_r) is the main key factor in this proposed algorithm. Generally, the basis concept of any BME algorithms is to compare the blocks and find

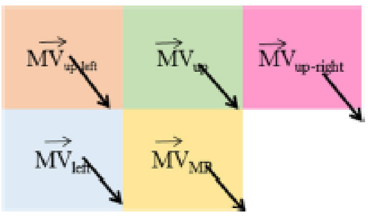


Fig. 1 Neighboring blocks details for median predictor

out the differences. So basically, distortion is already used in the BME algorithms. MDB is also based on the minimum differences.

The proposed algorithm uses this basic information to define the motion and there is no extra computational overhead in this logic. DS_r is calculated based on the starting distortion value (DS_p) and ending distortion value (DS_c) at each step. The degree of motion is predicted based on DS_r as mentioned in (1) and accordingly the search pattern is chosen in the proposed BME.

$$DS_r = DS_c/DS_p. \tag{1}$$

2.3 Threshold definition and motion categorization

Although, on the run time DS_r can be captured and use to classify motion. However, to determine whether the motion is small or large, a threshold parameter (TH_δ) is required. To define TH_δ testing was done with some of the benchmark video sequences, which are representative of different types of motion and also widely use in HEVC standard testing. *Johnny* (Class E), *BQSquare* (Class D), and *BasketballDrill* (Class C) are typically representatives of low, medium, and high motion respectively. The details of benchmark video sequences are summarized in Table 1. Intentionally, different class of video sequences with different resolutions, are taken into consideration to exclude the biasness of class and resolution in threshold calculation.

Table 1 Test video sequence description

Class	resolution	Sequence names	Frame rate (fps)	Description
A	2560 × 1600	Traffic	30	Complex content with medium motion
		People on street	30	Medium motion with rich texture
B	1920 × 1080	Basketball drive	50	High motion with medium motion
		Kimono	26	Medium motion with simple background
C	832 × 480	Party scene	50	Medium motion with zoom-in effect
		BQ mall	60	Medium motion with camera movement
D	416 × 240	BQ square	60	Medium motion with camera movement
		Basketball pass	50	High motion with rich texture
E	1280 × 720	Johnny	*60	Slow motion with static background
		Four people	60	High motion with rich texture

TZS with square search pattern of HEVC reference software (HM16.10) is used for the testing purpose. Starting search center distortion value (DS_p) and final MV distortion value (DS_c) are captured for these three standard video sequences over 15 frames for a particular quantization parameter (QP) value (37) and DS_r is calculated. DS_r is considered as random variables. Then according to the central limit theorem, the distribution of these independent random variables tend toward a normal distribution. From this distribution, the corresponding mean and standard deviation are calculated. These values are used to plot Gaussian distribution.

Figure 2 shows the combined representation distributions of three standard video sequences. Threshold 1 (TH_1) and Threshold 2 (TH_2) are obtained as shown in the Fig. 2. TH_1 and TH_2 help to define the motion degree. However, from the observations and intensive experiments, the real time video sequences contain motionless or very less motion in its major portion. Also, as per the study and previous reports, the arm length of the search pattern is not significantly different between large and medium motion video sequences. Thus, to reduce the computational complexity as well as encoding time, no separate pattern is considered for Medium motion in this proposal.

$$TH_\delta = 0.59. \tag{2}$$

In this proposal, the mean value of TH_1 and TH_2 is considered as TH_δ. So all the video sequences are classified into large or small motion categories based on TH_δ. After the extensive testing, TH_δ is defined as (2). When DS_r value is greater than the TH_δ, the motion is categorized as large motion, otherwise the motion is categorized as small motion. Based on this motion category, the search pattern is defined.

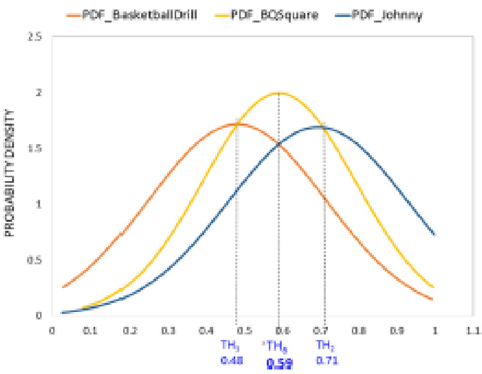
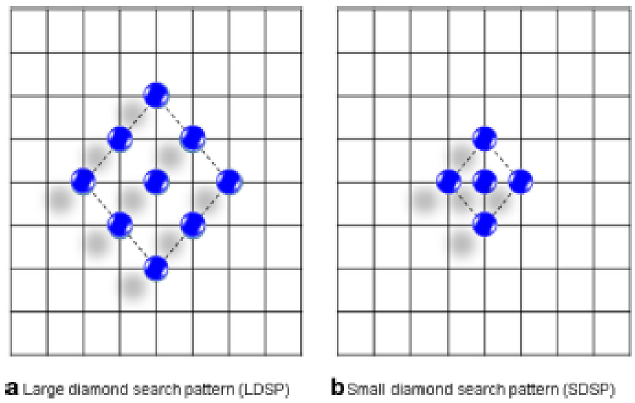


Fig. 2 Distortion ration distribution

Fig. 3 Two search patterns of the proposed algorithm



2.4 Pattern selection and search

During pattern selection, DS_p and DS_c are captured from both median MV or $(0, 0)$ positions. DS_r is achieved and compared with pre-defined TH_δ to categorize motion degree into small or large. SDSP for small motion and LDSP for large is considered. Figure 3 shows LDSP and SDSP pattern, which are used in the proposed algorithm.

LDSP composes of nine search points from which eight search points are surrounding the center. SDSP composes of five search points from which four are surrounded the center. The proposed algorithm evaluates all the search points to find out the MBD point, which determine the pattern and therefore switching of the patterns occur. The switching of both the search patterns for LDSP is demonstrated in the Figs. 4 and 5 when the MBD point found in the previous step at (a) one of the corner points, (b) one of the edge points, and (c) center point. If the motion is again categorized as large, LDSP is selected for next step. Instead of searching eight points, program will check additional three or five points to find out the next MBD point.

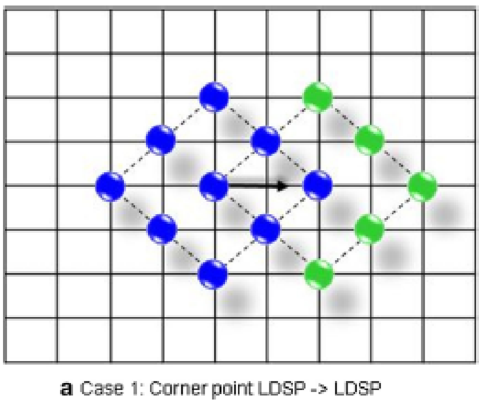


Fig. 4 Switching of both search patterns—LDSP to LDSP

Similarly, for small motion, instead of searching four points, program will check additional three or two points to find out the next MBD point. The switching of SDSP is demonstrated in the Fig. 6 when the MBD point is found in the previous step at (a) one of the corner points and (b) one of the edge points. And continue the search until MBD is found at center of the SDSP pattern.

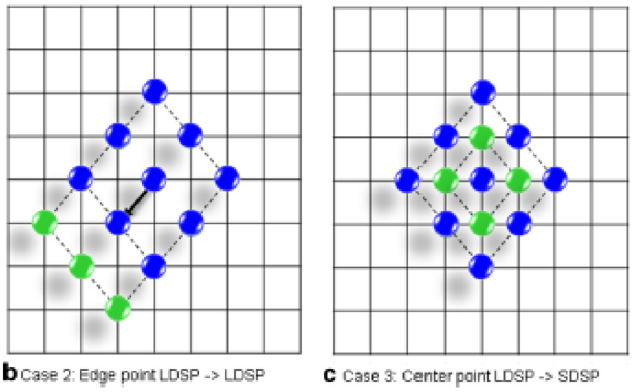
The details process flow of the propose algorithm is described in Fig. 7 and lets have close look into the process of the proposed algorithm.

- ☐ At the very beginning, median predictor is calculated based on the median MV value of the left, up, and up-right blocks of the current block as described in Fig. 1.
- ☐ Compare $\overrightarrow{MV}_{MP})$, $\overrightarrow{MV}_{\text{left}}$, $\overrightarrow{MV}_{\text{up}}$, $\overrightarrow{MV}_{\text{up-right}}$, and $\overrightarrow{MV}_{(0,0)}$. The minimum among these will be selected as the ISP.
- ☐ While selecting the ISC, DS_p, and DS_c are captured from both median MV or $(0, 0)$ positions. DS_r is achieved as described in (3) and compared with pre-defined TH_δ to classify the motion degree at beginning of the search, which imposes search pattern. This is the core idea of this proposed algorithm.

$$DS_r = DS_{(\text{median})}/DS_{(0,0)}. \tag{3}$$

- ☐ If the motion is classified as large, then LDSP (shown in Fig. 3) will be deployed at ISC. DS_p resets and stores

Fig. 5 Switching of both search patterns—LDSP to SDSP



the search center distortion information. All the eight points are evaluated using cost function [17].

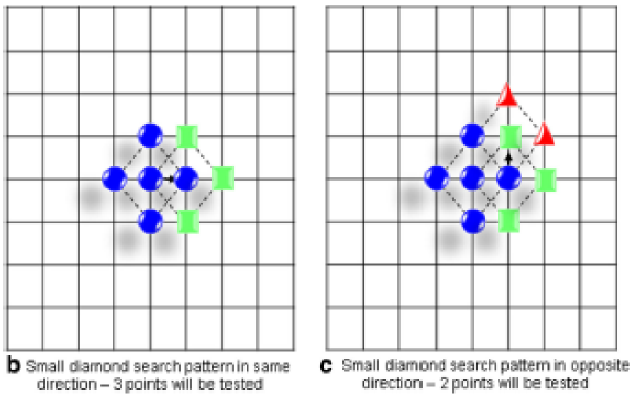
- Once MBD point is achieved, the center of the search pattern is changed to the MBD point (found in the previous step) and DS_c is captured from MBD point. DS_p at previous ISC and DS_c at new MBD point are used to derived the new DS_r, which in turn helps to select the search pattern again. The logic to get the DS_r from the 2^{nd} step onward is described in (4).

$$DS_r = DS_{(CurMBD)}/DS_{(PrevMBD)} \tag{4}$$

- If the motion is again classified as large, then LDSP is used based on the current MDB point and additional five or three points are checked as demonstrated in Figs. 4 and 5.
- Again, the distortion ratio are calculated with the previous and current MBD points as mentioned in (4). This process will continue until MBD is found at center of the search pattern.
- If the MBD is found at center of the LDSP, the SDSP pattern will be used as shown in the Fig. 5c.
- If the motion is classified as small, then SDSP is used. Four search points of SDSP are checked for MBD point. If the MBD point is at center of SDSP, then that will be the final MV for that search. Otherwise, again the DS_r is calculated as (4) and continue the search process until MBD is found at center of the SDSP.

Therefore, instead of using a fix pattern at staring, the proposed algorithm hypothesizes of employing diverse search patterns based on motion size in each and every steps until MBD is at center. With this logic, the proposed algorithm is able to reduce the encoding time to a great extend while maintaining the quality of the video. The experimental results are discussed in the next section.

Fig. 6 Traversal of small diamond pattern in the proposed algorithm



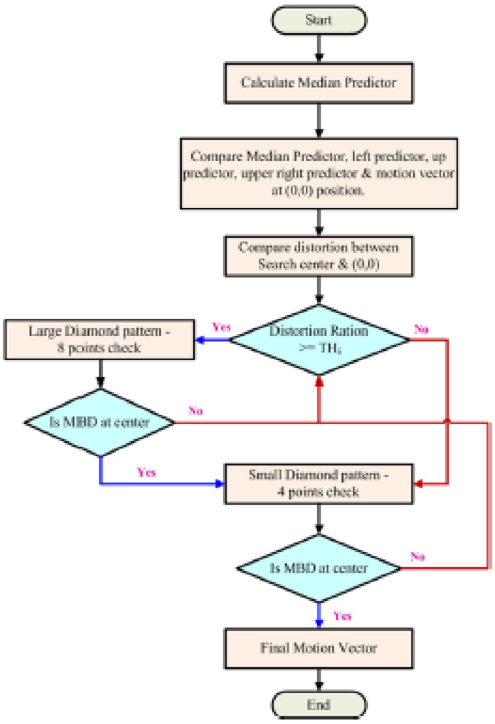


Fig. 7 Flow diagram of the proposed algorithm

3 Experimental results and discussions

The proposed algorithm was tested on HM 16.10 [18]. All simulation was performed on a PC with an Intel core i7-3770 processor with a 3.5 GHz clock speed and 8 GB of RAM running on Windows 10. While performing the testing below mentioned test conditions were maintained.

1. Configurations were followed as mentioned in HEVC test condition standard document [19].
2. Standard test video sequences with different classes and resolutions were chosen for experiment. The description of different test video sequences is given in Table 1.
3. According to the standard test conditions [19], Class E sequences were not used for RA-Main profile and Class A sequences were not used in LB-main profile testing.
4. Other important test parameters for test experiments are mentioned in the Table 2.

While doing the performance evaluation, the proposed algorithm was evaluated with TZS algorithm and DS [9] using HEVC reference software (HM) version 16.10. Required time to encode video sequence, number of bits

Table 2 Other test parameters

Max. CU size	64×64
Max. CU depth	4
QP	22, 27, 32, 37
Search range	$[-64, 64]$
Coding profile	Low delay main (LB-main), Random access (RA-main)

in the encoded bit-stream, and peak signal-to-noise ration (PSNR) are three parameters taken into consideration during experiment. ΔBit means the total bit rate changed (in percentage) as represented in (5). ΔY-PSNR means the change of Y-PSNR as summarized in (6) and ΔT denotes amount of reduced time, calculated by (7). Bjontefaard Delta PSNR (BDPSNR) and Bjontegaard Delta BR (BDBR) are the average PSNR differences in dB for the same bit-rate along with the average BD differences in percent for the same PSNR receptively and computed accordingly (refer [20]). All these measurements were used for total 50 frames of test sequences, mentioned in Table 1.

$$\Delta Bit = \frac{\mathrm{Bitrate}_{\mathrm{proposed}} - \mathrm{Bitrate}_{\mathrm{original}}}{\mathrm{Bitrate}_{\mathrm{original}}} \times 100\%, \tag{5}$$

$$\Delta \mathrm{PSNR} = \mathrm{PSNR}_{\mathrm{proposed}} - \mathrm{PSNR}_{\mathrm{original}}, \tag{6}$$

$$\Delta T = \frac{\mathrm{Time}_{\mathrm{proposed}} - \mathrm{Time}_{\mathrm{original}}}{\mathrm{Time}_{\mathrm{original}}} \times 100\%. \tag{7}$$

3.1 Performance comparison in LB-main profile

Table 3 shows the performance comparison of the proposed algorithm with respect to TZS, as well as with DS [9] for LB-main profile. On average, around 42.56, 43.63, 36.95, and 37.59% time deduction for class B, C, D, and E video sequences was achieved respectively in compare with original HM software, while 38.12, 40.32, 42.55, and 40.37% time deduction was obtained compare with original DS algorithm [9].

For “Kimono” video sequence, due to its simple background, encoding time deduction was observed approximately 43.16% with only 0.011 dB deduction in PNSR while 0.05% increased in bitrate. Also, 0.012 dB increased for BDPSNR and 0.36% increased in BDbitrate. “BasketBall Drive” video sequence is classified as large motion video sequences and the proposed algorithm achieved 41.95% encoding time deduction with 0.005 dB deduction of PSNR for this large motion case. Also, “Johnny” video sequence contains small motion with static background and gave 42.85% of time deduction with negligible deduction of video quality around 0.001 dB comparing with TZS.

Table 3 The overall comparisonal performance of the proposed algorithm with TZS and DS [9] in HEVC reference software (HM) 16.10 for low delay main (LB-main) profile

Sequence	QP	TZS ΔBit (%)	TZS ΔY-PSNR	TZS ΔT (%)	TZS BDPSNR (dB)	TZS BDBR (%)	Diamond search [9] ΔBit (%)	Diamond search [9] ΔY–PSNR	Diamond search [9] ΔT (%)	Diamond search [9] BDPSNR (dB)	Diamond search [9] BDBR (%)
Basketball drive	22	0.238	0.001	−42.969	−0.021	0.915	−0.171	−0.001	−38.968	0.004	−0.183
	27	0.492	−0.004	−40.308			−0.225	−0.002	−35.797		
	32	0.969	−0.003	−41.985			−0.399	−0.006	−36.807		
	37	1.073	−0.015	−42.557			−0.249	0.001	−37.976		
Kimono	22	0.046	−0.001	−42.701	−0.0128	0.369	−0.171	−0.002	−38.446	0.006	−0.169
	27	0.119	−0.003	−43.218			−0.112	0.001	−40.217		
	32	0.079	−0.015	−43.321			−0.158	0.003	−38.561		
	37	−0.017	−0.027	−43.437			−0.378	−0.009	−38.249		
Party scene	22	0.013	0.000	−40.749	−0.012	0.248	−0.113	0.002	−38.547	−0.001	0.021
	27	0.212	0.001	−44.322			0.027	−0.003	−39.037		
	32	0.348	−0.003	−44.242			−0.006	−0.003	−38.462		
	37	−0.128	−0.016	−43.917			−0.123	−0.002	−39.098		
BQ mall	22	0.259	0.003	−44.680	−0.026	0.627	−0.097	0.001	−41.205	0.007	−0.156
	27	0.344	0.002	−44.577			−0.226	0.005	−41.392		
	32	0.877	−0.005	−44.207			0.144	−0.004	−42.252		
	37	0.749	−0.012	−42.372			−0.384	−0.007	−40.603		
Basketball pass	22	−0.044	−0.002	−36.101	−0.003	0.087	−0.080	0.009	−40.784	0.005	−0.103
	27	0.134	−0.005	−37.180			−0.235	−0.002	−41.885		
	32	−0.324	−0.002	−37.624			0.159	0.007	−42.007		
	37	0.492	−0.006	−38.018			0.434	0.016	−41.736		
BQ square	22	0.190	−0.001	−35.702	−0.005	0.101	0.139	0.004	−43.093	0.004	−0.100
	27	0.032	0.011	−36.516			−0.072	0.011	−43.187		
	32	0.035	−0.022	−35.155			0.026	−0.009	−42.378		
	37	−0.196	−0.011	−38.302			−0.142	0.005	−43.659		
Four people	22	0.374	−0.005	−41.369	−0.011	0.412	0.230	0.003	−40.573	−0.001	0.106
	27	0.321	0.004	−29.067			0.176	0.006	−42.420		
	32	0.471	−0.005	−33.085			0.245	−0.001	−41.331		
	37	0.074	−0.005	−25.843			−0.242	−0.005	−41.652		
Johnny	22	−0.166	0.000	−40.432	0.000	0.161	−0.099	0.000	−40.622	0.005	−0264
	27	0.126	0.001	−43.533			−0.070	0.003	−36.293		
	32	0.225	−0.009	−43.042			−0.424	−0.002	−37.673		
	37	−0.275	0.004	−44.403			0.309	0.016	−42.413		
Average (%)		**0.223**	**−0.005**	**−40.550**	**−0.011**	**0.365**	**−0.071**	**0.001**	**−40.346**	**0.004**	**−0.106**

Items in bold face represented the average performance of the each algorithm

Based on center-biased characteristic of motion vectors, for stationary and slow motion video sequences, the best motion vector of each prediction block (PB) is expected to be near from the center. Using adaptive pattern LDSP or SDSP, it is very beneficial to handle both type of motion sequences. The performance of the proposed algorithm was best for slow motion sequences, while it could significantly performs for medium or high motion video sequences too like medium motion including "BQ Mall" and "Party Scene" with zoom-in effect. Around 43% encoding time reduction was observed for class C.

On the other hand, only 32.34% of time deduction has been observed in "FourPeople" video sequence which contains high motion with complex rich texture background. However, the quality of the video was maintained with very marginal PSNR degradation of 0.003 dB. Proposed algorithm performs little low in respect to deduction of encoding time for high motion videos, improvement of the proposed algorithm for this exceptional cases are scope of future endeavor. In general, the overall performance of the proposed algorithm was good, approximately 40.18% encoding time reduction with very marginal PNSR degradation of 0.005dB.

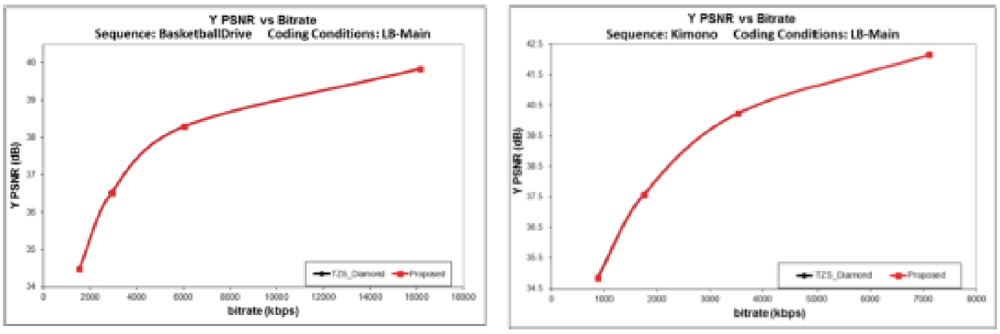


Fig. 8 Rate-distortion (RD) curves for *Basketball Drive* and *Kimono* sequences for class B in low delay main (LB-main) profile in comparison with TZS

Table 4 The overall comparisonal performance of the proposed algorithm with TZS and DS [9] in HEVC reference software (HM) 16.10 for random access (RA-main) profile

		TZS					Diamond search [9]				
Sequence	QP	ΔBit (%)	ΔY-PSNR	ΔT (%)	BDPSNR (dB)	BDBR (%)	ΔBit (%)	ΔY-PSNR	ΔT (%)	BDPSNR (dB)	BDBR (%)
Traffic	22	−0.037	−0.007	−33.394	−0.031	0.908	0.744	−0.002	−31.947	−0.051	1.417
	27	−0.142	−0.020	−34.084			1.428	−0.004	−34.288		
	32	0.088	−0.050	−34.780			1.353	−0.006	−33.286		
	37	0.049	−0.071	−23.241			1.297	−0.004	−33.621		
People on street	22	0.296	−0.004	−38.055	−0.044	1.00	−0.342	−0.004	−35.549	0.24	−0.52
	27	0.098	−0.008	−39.240			−0.307	0.003	−36.448		
	32	0.892	−0.034	−40.980			−0.508	0.005	−39.053		
	37	−0.682	−0.080	−36.509			−0.844	0.012	−35.659		
	22	−0.190	0.009	−38.020	−0.033	1.64	0.182	0.000	−36.112	0.029	−1.141
Basketball drive	27	−0.157	−0.010	−35.191			−0.718	0.011	−33.928		
	32	0.371	−0.057	−37.465			0.106	0.032	−38.099		
	37	0.278	−0.115	−35.651			0.443	0.098	−36.719		
Kimono	22	0.163	0.009	−35.825	−0.006	0.179	0.960	0.001	−33.887	−0.040	1.150
	27	0.172	−0.005	−36.259			1.160	−0.005	−34.929		
	32	−0.361	−0.013	−35.365			1.308	0.006	−34.696		
	37	−0.614	−0.082	−33.186			0.817	−0.001	−32.737		
Party scene	22	0.242	−0.002	−37.236	−0.042	0.876	−0.161	−0.003	−35.689	0.007	−0.153
	27	0.482	−0.002	−38.436			−0.124	0.000	−36.582		
	32	1.069	−0.013	−38.361			0.005	0.006	−36.469		
	37	0.891	−0.029	−37.369			−0.539	−0.006	−35.738		
BQ mall	22	0.158	0.005	−36.487	−0.001	0.072	0.014	0.000	−34.714	−0.016	0.355
	27	−0.975	−0.013	−36.691			0.671	0.001	−35.308		
	32	0.126	−0.018	−34.436			0.437	0.014	−32.436		
	37	0.446	−0.018	−35.604			0.605	0.002	−34.443		
	22	0.401	−0.029	−36.789	−0.065	1.372	0.009	−0.003	−35.817	0.007	−0.127
Basketball pass	27	0.566	−0.031	−36.717			−0.493	−0.031	−35.683		
	32	0.874	−0.023	−37.891			−0.139	0.015	−35.502		
	37	1.985	−0.024	−36.346			0.123	0.019	−36.378		
BQ square	22	−0.160	0.001	−30.008	−0.002	0.052	−0.094	0.004	−37.037	0.002	−0.038
	27	0.348	0.012	−38.387			0.081	0.000	−36.424		
	32	−0.057	−0.006	−37.262			0.114	0.008	−37.425		
	37	−0.125	−0.015	−38.531			−0.516	−0.015	−36.913		
Average (%)		**0.246**	**−0.023**	**−36.056**	**−0.028**	**0.763**	**−0.221**	**0.005**	**−35.422**	**−0.005**	**0.117**

Items in bold face represented the average performance of the each algorithm

For class D video sequences, e.g., "BQ square" and "Basketball pass" (416 × 240), the proposed algorithm achieved 36% encoding time deduction with maintained video quality, only 0.002 dB loss in PSNR, and 0.31% increment in total bits. From the observation, it is concluded that the proposed algorithm achieves 38.55% encoding time deduction with a loss of 0.005 dB in PSNR (i.e., video quality) and 0.22% increment in total bits.

For performance comparison from a low to high bit rate, rate-distortion (RD) performances were captured under standard HEVC test conditions [19] for all video sequences as mentioned in Table 1. As HEVC is mainly focused on high resolution video sequences, only class B, the RD performances are shown in Fig. 8. The proposed algorithm performs very similar to the original HM 16.10 software, which results in negligible loss of quality and bit-rate.

3.2 Performance comparison in RA-main profile

The proposed algorithm had also been tested with RA-main profile, and approximately 36.05 and −35.42% encoding time deduction was observed with compare to both the algorithms. The details are shown in Table 4. On average, 35.87, 36.82, 36.49, and 35.03% of time deduction for class B, C, D, and A video sequences were achieved respectively compared with the original HM software and 35.13, 35.17, 36.39, and 34.98% time deduction in compare with original Diamond Search algorithm [9]. For "Kimono" video sequence, due to its simple background (resolution 1920 × 1080), encoding time deduction was observed 34.06% with only 0 dB deduction in PNSR while 1.06% increased in bitrate. For "BQ Mall" 34.22% and for "Party scene" 36.11% deduction of encoding time were achieved.

4 Conclusions

This paper has proposed a fast and robust block-based motion estimation (BME) algorithm for multimedia IoT platform to reduce the encoding time in HEVC. In order to make a fast BME algorithm, adaptive pattern selection approach has been taken into consideration based on motion degree.

For the determination of motion degree, the basic information of videos including distortion was taken into account. Based on the distortion ratio, the motion was categorized into either small or large. Therefore, instead of using a fix patterns at staring, the hypothesis of employing diverse search pattern based on motion size in each and every steps until MBD was at center of SDSP, results the algorithm with significant reduction in computation time while maintaining the quality.

A comparative study has been included with TZS algorithm of HEVC reference software and DS [9] in order to validate the applicability of the proposed algorithm to improve the standard. The results indicated the hypothesis of using of adaptive pattern from the beginning of search worked well in fast ME.

Few of the future prospective are summarized as follows: (a) the incorporation of parallel processing can open up new direction of research; (b) with the inclusion of directional information about motion degree, asymmetric search pattern can also be used instead of symmetric patterns to further reduce the encoding time; and (c) embedded processor aspect re-design and implementation based on graphic processing unit (GPU) parallel processing and single instruction and multiple data (SIMD) is needed because the environment of IoT has usually low computing power, slow communication speed, and short wireless communication range.

Acknowledgements This research was supported by the Sookmyung Women's University Research Grants (1-1703-2008).

References

1. Atzori L, Iera A, Morabito G (2010) The internet of things: A survey. Comput Netw 54:2787–2805
2. Ericsson (2013) More than 50 billion connected devices. Technical report, Ericsson, Stockholm
3. Wiegand T, Sullivan GJ, Bjontegaard G, Luthra A (2003) Overview of the h.264/AVC video coding standard. IEEE Trans Circuits Syst Video Technol 13(7):560–576
4. Bross B, Han WJ, Ohm JR, Sullivan GJ, Wang YK, Wiegand T (2013) High efficiency video coding (HEVC) text specification draft 10. In: Joint collaborative team on video coding (JCTVC) of ITU-T VCEG and ISO/IEC MPEG, Geneva, CH, JCTVC-L1003
5. Jain JR, Jain AK (1981) Displacement measurement and its application in interframe image coding. IEEE Trans Commun 29(12):1799–1808
6. Koga T, Iinuma K, Hirano A, Iijima Y, Ishiguro T (1981) Motion-compensated interframe coding for video conferencing. Proc. Nat. Telecom. Conf. New Orleans, LA G5.3.1G.5.3.5
7. Li R, Zeng B, Liou ML (1994) A new three-step search algorithm for block motion estimation. IEEE Trans Circuits Syst Video Technol 4(4):432–442
8. Po L-M, Ma W-C (1996) A novel four-step search algorithm for fast block motion estimation. IEEE Trans Circuits Syst Video Technol 6(3):313–317
9. Zhu S, Ma K-K (2000) A new diamond search algorithm for fast block-matching motion estimation. IEEE Trans Image Process 9(2):287–290
10. Cheung C-H, Po L-M (2002) A novel cross-diamond search algorithm for fast block motion estimation. IEEE Trans Circuits Syst Video Technol 12(12):1168–1177
11. Lam C-W, Po L-M, Cheung CH (2003) A new cross-diamond search algorithm for fast block matching motion estimation. Proc Intern Conf Neural Netw Signal Process 2:1262–1265
12. Cheung CH, Po LM (2002) A novel smallcrossdiamond search algorithm for fast video coding and videoconferencing applications. In: Proceedings IEEE international conference on image processing, pp 681–684

13. Yun C, Dai K, Wang Z, Guo J (2005) A fast motion estimation algorithm based on diamond and simplified square search patterns. In: Iberoamerican congress on pattern recognition, pp 440–449
14. Shen-Chuan T, Chen Y-R, Chen Y-H (2007) Small-diamond-based search algorithm for fast block motion estimation. Signal Process Image Commun 22(10):877–890
15. Djoudi K, Belbachir MF (2015) Star diamond: an efficient algorithm for fast block matching motion estimation in h264/AVC video codec. In: Multimedia tools and applications, pp 1–15
16. Zhu C, Lin X, Chau L-P (2002) Hexagon-based search pattern for fast block motion estimation. IEEE Trans Circuits Syst Video Technol 12(5):349–355
17. Ohm J-R, 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
18. HEVC Test Model (HM), HEVC Reference software [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/
19. Bossen F (2012) Common test conditions and software reference configurations. In: JCTVC-i1100, JCT-VC document 9th Meeting, Geneva
20. Bjøntegaard G (2001) Calculation of Average PSNR Differences between RD curves. In: ITU-T SG16/Q6, 13th VCEG Meeting, Austin, Texas, USA, Doc. VCEG-M33