IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 7 Issue 1, January 2020
ISSN (Online) 2348 – 7968 | Impact Factor (2019) – 6.248
www.ijiset.com

Diamond Search Improvement for Motion Estimation in High Efficiency Video Coding

Dr. Eng. Jamal Kamil Al-Rudaini

Computer Engineering Techniques Department, Al-Nisour University College
Baghdad, Iraq

Abstract
High Efficiency Video Coding (HEVC) for video stream is illustrated in this work. Motion estimation stage of the encoder is focused. The main part of motion estimation (ME) is the search algorithm selection. Improvement of coding efficiency for video sequences using Diamond Search algorithm compared with full search algorithm is the main task of this work. Diamond Search algorithm resulting in faster search speed compared with other algorithms. The experimental results are done on different sequences for comparison. The procedure was showing high efficiency with effective and robust results.
Keywords: Motion compensation, Motion estimation, High Efficiency Video Coding.

1. Introduction

The advantage of Advanced Video Coding (AVC) which known as H.264 standard was most likely coded way for video recording in webs and mobile watching. This standard was developed in 2003 and become widely used in 2008 [1].
HEVC is recognized also as H.265. HEVC has been adopted to be used in 2013 for video compression by the group of Moving Picture Experts Group (MPEG). HEVC shows high ability as twice efficiently as AVC for video compression of higher quality level. Also, HEVC has efficient recording and distribution of video compared with Ultra High Definition (UHD) [2].
Adaptive Motion Vector Predictor mode (AMVP) and merge mode are the main two motion vectors (MV) applied in HEVC. AMVP data is coming from the blocks of the reference picture and the previously predicted blocks. The merge mode takes the data from neighboring motion information. The most time consumption occurs during the motion estimation (ME) stage of the encoder during searching the motion vectors. Reduction of mathematical complexity of motion estimation algorithms in the video coding attracts many researches. A suggested procedure is presented in this work to decrease the time during the motion estimation stage of the encoder without decreasing the coding efficiency.

2. Coding Techniques Overview

H.265 is a new video compression standard. It is developed by the Joint Collaborative Team on Video Coding (JCT-VC). The JCT-VC attracts image and video encoding experts around the world. HEVC is approved by Motion Picture Experts Group (MPEG) and Video Coding Experts Group (VCEG) [3].
The goal of HEVC development is to provide a compression efficiency double of the AVC standard. HEVC enables video is compressed to half the size of AVC file and it delivers significantly a better visual quality. The efficiency of HEVC coding of video files compared with AVC standard is shown in the following facts [4]:
Motion compensated prediction technique is the most powerful stage of video compression standards. Each frame of video stream is presented by Blocks of pixels. Each block is encoded with reference to another area in the same frame as (intra-prediction), or in another frame as (inter-prediction). The decoded block in H.264/AVC is up to 16×16 pixels where in HEVC could be up to 64 x 64 pixels.
Different block sizes are allowable in HEVC coding for predicted blocks to be coded in prediction quad-tree. The residual error obtained from a 32×32 inter coded coding unit is combined by a mixture transforms as 16×16, 8×8, and 4×4 blocks.
Encoded motion vectors in HEVC has 35 intra-picture directions while only 9 for AVC.
Adaptive Motion Vector Prediction method (AMVP) used in HEVC gives extra improvement to HEVC for inter-prediction.
Reducing artifacts in HEVC is another advantage of this coding by using of de-blocking filter.

3. Searching Algorithms:

The searching algorithm in motion estimation and motion compensation is an important manner in video coding techniques [5]. The basic type of searching methods is the full-search algorithm. The computational time is very high and takes more than 96% of the total encoding period [6, 7]. Diamond-search algorithm is presented and tested in this work to decrease the computational time without affecting the coding efficiency.

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. The full search procedure uses a 64×64 block to compare all blocks in the search range. Sum of absolute differences (SAD) is the most commonly used for matching criterion. SAD is chosen for its simplicity and easy hardware implementation.

3.2 Diamond-Search Algorithm:

The Diamond 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 step, increasing the search distance from 1 to 8 and searching from 1 to 64 of the search areas [1]. After the search, the point with the smallest Sum of Absolute Difference value is set as the midpoint of the next search step, and the search distance is stored (Fig. 1).

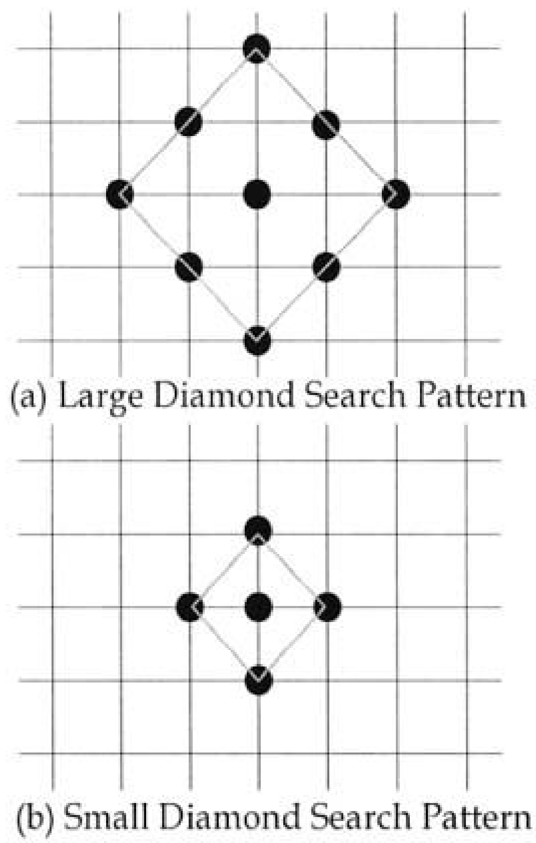

Fig. 1: The 8 and 4 points diamond search algorithms styles.

4. Experiments and Results

In this paper, a comparison study between HEVC with DS algorithm and HEVC with FS algorithm that allow reducing time have been presented. The performed experiments showed that the HEVC with DS algorithm led to average time reduction to over 26% in random access configuration and 29% in low delay configuration for video sequences recorded with the HEVC with DS algorithm comparing to the HEVC with FS algorithm. In same time, during use of both algorithms the video performance doesn't change as shown in the selected frames in (Fig. 2).
The proposed method can estimate the motion between consecutive frames with the shortest time. Spacious tests have been executed in order to assess the performance of proposed method. Evolutions have been performed in reference of Two Dimensional- High Efficiency Video Coding 2D-HEVC codec version HM-KTA [8]. Each experiment has been performed accordance with the Common Test Conditions (CTC) of Joint Collaborative Team on Video Coding JCT-VC, which dub encoder configuration for valuation of 2D codec execution used by International Organization for Standardization (ISO). Main parameters of the encoders have been collected in Table 1. The coding efficiency is measured at four operating points (QP = 27; 22; 37; 32), calculating time reduction and the average rate reduction in comparison to HEVC with full search algorithm (FS) following the method described in [9] and the results can be summarized in Table 2. The Peak Signal to Noise Ratio (PSNR) values and the corresponding total bitrates for all test sequences for HEVC with diamond search algorithm and HEVC with full search algorithm and presented in Table 3.

Table 1: Main parameters configuration of the encoders.

Parameter	**Value**
Profile	Main
GOP size	4
FrameSkip	0
Intra period	24
SEIDecodedPictureHash	1
MaxCUHeight	256
MaxCUWidth	256
QuadtreeTULog2MaxSize	6
MaxPartitionDepth	6
LoopFilterTcOffset_div2	-2
InternalBitDepth	8
SearchRange	64

Table 2: BD-rate using Bjontegaard by mean of DS algorithm verse FS algorithm.

Sequences	Random Access Configuration			
	BD-Rate			Encoder time
	Y	U	V	
BasketballDrill_832x480	0.40%	-0.59%	0.31%	20.43%
BQSquare_416x240	-0.20%	0.41%	-0.20%	34.92%
BlowingBubbles_416x240	0.78%	-0.11%	0.33%	22.72%

Sequences	Low Delay Configuration			
	BD-Rate			Encoder time
	Y	U	V	
BasketballDrill_832x480	0.21%	0.84%	1.22%	42.88%
BQSquare_416x240	-0.19%	-0.48%	-0.65%	24.51%
BlowingBubbles_416x240	0.56%	1.19%	2.06%	20.50%

Reference:

[1] Iain E. Richardson , “The H.264 Advanced Video Compression Standard”, 2nd Edition, John Wiley & Sons 2010.

[2] Benny Bing, “Next-Generation Video Coding and Streaming”, John Wiley & Sons, 2015.

[3] Sullivan, G.J., Ohm, J.R., Han, W.J., et al: “Overview of the high efficiency video coding (HEVC) standard”, IEEE Trans. Circuits Syst. Video Technol., 22, 2012, pp. 1649–1668.

[4] Kanika Sihag, C. Lamba, “Algorithm and Architecture Design of High Efficiency Video Coding (HEVC) Standard", IJCSMC, Vol. 5, Issue. 10, October 2016, pp.171 – 178.

[5] J. Abbas, A. Alanssari, M Patlayenko, V. Pilyavskiy, "Improvement to Motion Estimation for High-Efficiency Video Coding" of the O.S. Popov ONAT 2019 (1), pp. 112-120.

[6] Fatma E. Sayadi, Marwa CHouchene , Haithem Bahri , Olfa Haggui, Bouraoui Ounir, “Improved approach for full search motion estimation on GPU”, International Journal of Computers, Volume 2, 2017, pp. 220-222.

[7] Ankita S. Sawalkar, Vinay Keswani, “SVM Based Saliency Map Technique For Reducing Time Complexity In HEVC”, International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 06, June - 2017, pp. 880-882.

[8] Joint Collaborative Team on Video Coding, “HMKTA reference software 16.6”.

[9] G. Bjontegaard, “Calculation of Average PSNR Differences between RD-curves,” ITU-T SG16, Doc. VCEG-M33, Austin, USA, Apr. 2001.

ISSN (Online) 2348 – 7968 | Impact Factor (2019) – 6.248

www.ijiset.com

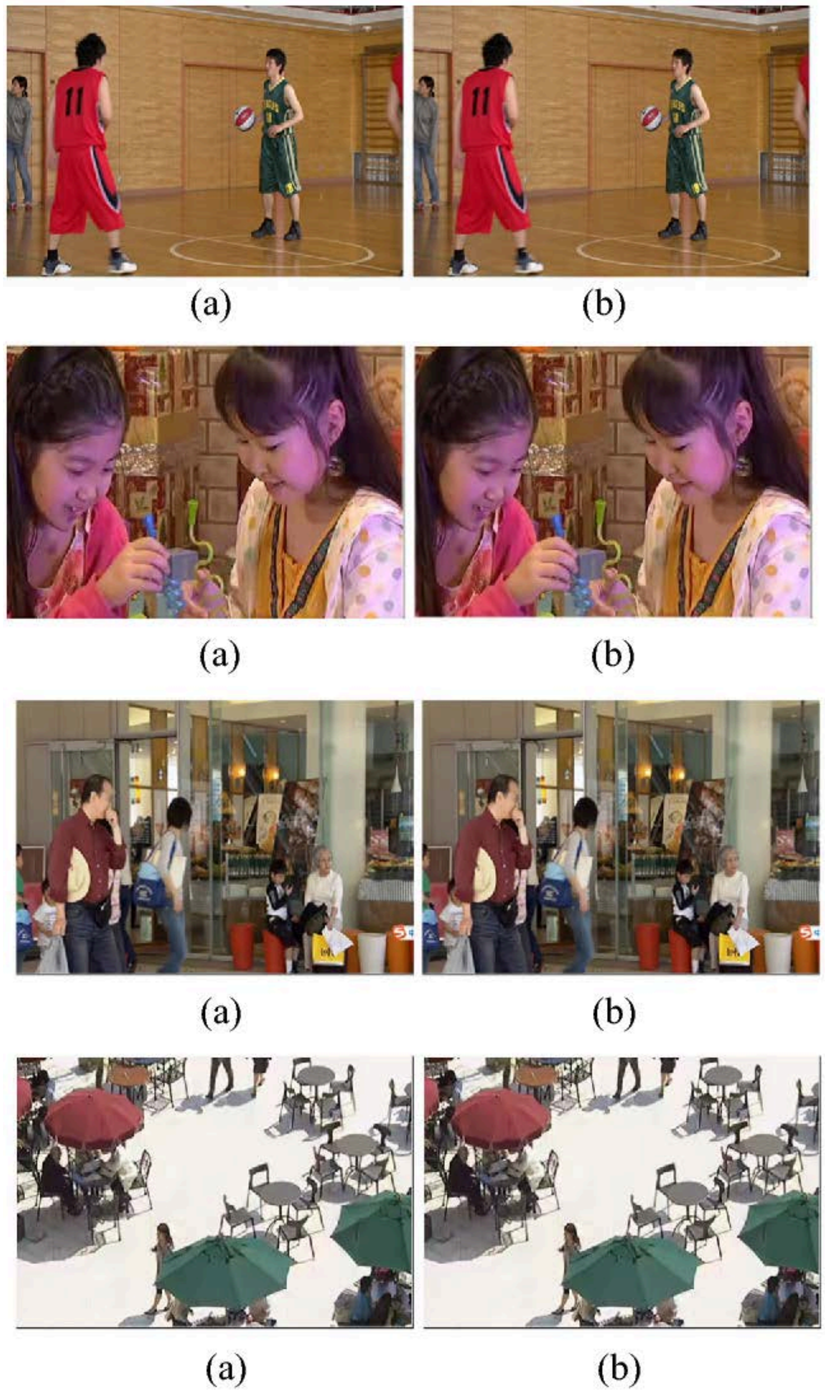

Fig. 2: video sequences recorded with HEVC using:
DS algorithm (a) comparing with FS algorithm (b).

Table 3: Experimental results for all test sequences for DS algorithm and FS algorithm

Sequences	QP	Random Access Configuration									
		Diamond Search					Full Search				
		Bitrate [kbit/s]	YPSNR [dB]	UPSNR [dB]	VPSNR [dB]	Time [sec]	Bitrate [kbit/s]	YPSNR [dB]	UPSNR [dB]	VPSNR [dB]	Time [sec]
BasketballDrill	22	3196.4	41.1077	43.5739	44.4444	1616.757	3184.98	41.101	43.5694	44.43	7102.03
	27	1542.36	38.075	41.4353	41.9611	1508.909	1539.3	38.0775	41.3984	41.9411	6966.588
	32	760.78	35.1153	39.6344	39.9609	1197.042	763	35.1514	39.6297	39.9861	6370.892
	37	408.04	32.6144	38.3901	38.4879	1129.47	407.96	32.6377	38.4022	38.57	6101.787
BQSquare	22	1748.736	39.5422	43.9614	44.8777	324.781	1757.736	39.5566	43.9663	44.8763	673.482
	27	864.936	36.0812	41.6859	42.4362	184.159	865.536	36.1061	41.7053	42.4392	618.329
	32	461.88	32.8423	39.8673	40.7357	164.714	460.608	32.8352	39.875	40.706	541.297
	37	263.448	29.8783	38.6893	39.437	158.611	263.328	29.8785	38.699	39.4704	507.478
BlowingBubbles	22	1664.32	38.834	41.4796	43.5813	410.681	1663.16	38.8373	41.4579	43.544	1633.624
	27	771.24	35.6354	39.0043	41.0389	281.797	769.42	35.6644	38.9681	40.9955	1291.776
	32	372.5	32.5958	37.1243	39.0134	277.237	371.98	32.6295	37.1511	39.0727	1321.329
	37	183.62	29.8768	35.6457	37.5903	258.593	183.2	29.8825	35.6555	37.6471	1127.995

equences	QP	Low Delay Configuration									
		Diamond Search					Full Search				
		Bitrate [kbit/s]	YPSNR [dB]	UPSNR [dB]	VPSNR [dB]	Time [sec]	Bitrate [kbit/s]	YPSNR [dB]	UPSNR [dB]	VPSNR [dB]	Time [sec]
BasketballDrill	22	3497.66	41.0595	43.3605	44.2485	7932.028	3491.96	41.0572	43.3804	44.2579	11475.57
	27	1660.3	37.9549	41.1426	41.776	7881.465	1663.34	37.9636	41.1752	41.8116	11094.21
	32	807.12	35.0039	39.328	39.7498	1449.399	807.9	35.0237	39.351	39.8061	8095.329
	37	425.9	32.4807	37.9878	38.2913	1075.24	424.18	32.4812	37.9769	38.2998	7992.777
BQSquare	22	2184.6	39.2962	43.4564	44.3278	294.771	2182.896	39.3087	43.4662	44.3155	1058.279
	27	1005.864	35.4139	41.2621	42.0951	213.68	1006.896	35.4286	41.2535	42.0741	926.035
	32	499.392	32.0702	39.5196	40.2651	188.02	499.056	32.0737	39.4937	40.2518	886.76
	37	270.096	29.1056	38.5223	39.1443	169.733	271.32	29.1324	38.5281	39.1451	655.367
BlowingBubbles	22	1973.52	38.7527	40.9916	42.9127	686.747	1971.18	38.741	40.9786	42.8821	2327.113
	27	864.68	35.2021	38.4334	40.4727	452.965	861.36	35.2076	38.4714	40.5597	2186.048
	32	399.14	32.0557	36.6307	38.5566	343.068	393.94	32.0353	36.6296	38.5635	2059.792
	37	187.6	29.2556	35.2226	37.1028	286.302	186.48	29.2679	35.2196	37.1097	1894.602