Fast Motion Estimation for HEVC Video Coding

Randa Khemiri*[1], Nejmeddine Bahri[2], Fatma Belghith[2], Fatma Ezahra Sayadi[1],
Mohamed Atri[1], Nouri Masmoudi[2]

[1]Electronics and Microelectronics Laboratory,
University of Monastir, Environment Street, 5019 Monastir,
Tunisia
E-mail. Randa.khemiri@gmail.com

[2]Electronics and Information Technology Laboratory,
University of Sfax, National Engineering School of Sfax,
Tunisia

Abstract—**In this paper, a fast configuration for Motion Estimation (ME) is described in order to reduce the computational time of the new High Efficient Video Coding (HEVC). This configuration uses the Coded Block Flag (CBF) Fast Method (CFM), the Early Coding Unit (CU) termination (ECU) and the Early Skip Detection (ESD) modes. The Diamond Pattern is used as a search algorithm for ME in the encoding process. Compared to the latest original reference software test model (HM) 16.2 of the HEVC, experimental results had showed that the complexity is reduced, in average, by 56.75% with a small bit-rate and PSNR degradation.**

Keywords-HEVC; HM16.2; Fast configuration (ECU, ESD, CFM); Motion Estimation; Diamond search pattern

I. Introduction

Nowadays, the proposed HEVC [1] is the most recent standard used on video compression [2]. When compared to the previous standards, it can achieve as equivalent visual quality as H.264/MPEG-4 AVC encoders, by only using 50% less bit-rate [3]. In HEVC, the motion estimation is the most important part of video compression occupying around 70% of the total encoding time [4]. It is the major factor in the compression efficiency as it occupies a large part of the whole coding arithmetic. In order to reduce the encoding time, several researchers studied ME-algorithm to optimize the test zonal search (TZ search). Purnachand et al. [5] replaced the 8-point diamond and 8-point square with a 6-point hexagonal in the TZS algorithm. That had reduced the encoding time by almost 50% compared to TZS algorithm with the same PSNR and bit-rate. In addition, the determination of the most optimal CU partition for each Coding Tree Unit (CTU) and the best Prediction Unit (PU) mode for each CU causes an increase in computational complexity. In the literature, many efforts had been focusing on exploring algorithms established on fast decision. Liquan et al. [6] had presented an algorithm based on skipping some specific depths which are rarely used in the previous frame. The proposed algorithm reduces 26% in the execution time with a slight increase in bit-rate as well as a negligible coding efficiency loss.

This paper adopts some options for fast mode decision used for CU partitioning. The rest of this paper is organized as follows: In Section 2, an overview of the TZ Search algorithm, which presents the motion estimation algorithm, is provided. Section 3 presents the proposed configuration for the Random-Access-main profile (RA) to reduce the execution time. In Section 4 a comparison between this algorithm and the original one (HEVC reference software HM16.2) [7] is discussed. Finally, a conclusion and some future works are presented in Section 5.

II. TZ Search Algorithm

The TZSearch is the algorithm used in the ME process in HEVC. This algorithm is divided into four distinct main steps: The PMV (Prediction Motion Vector), the first research using the diamond or the square pattern, the raster search and the refinement. These steps are summarized as follows [8, 9]:

A. Prediction Motion Vector(PMV)

The TZS algorithm employs four predictors: the left predictor, the up, and the upper right ones to compute the median predictor of the corresponding block.

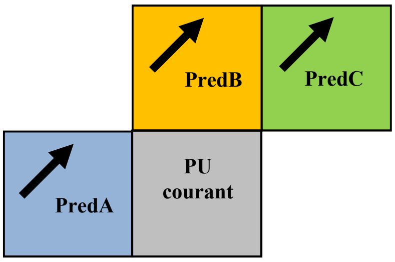


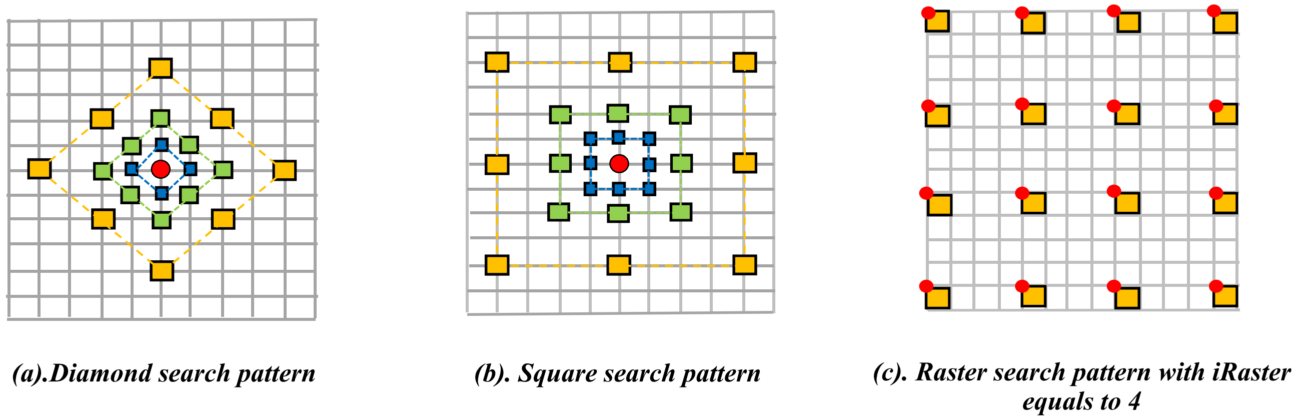
Figure 1. MV adjacent a current PU

The median is computed using the Eq. 1: Median(A, B, C) = A + B + C − Min(A, Min(B, C)) − Max(A, Max(B, C)). This is the starting point for the next step.

B. Initial Grid Search

After selecting the start position, the second step is to perform the first search and to determine the search range and the search pattern. This step consists of finding the search window using a diamond or square pattern as shown in Figure 2(a) and Figure 2(b) with distances ranging from one to "searchrange" (For HM16.2 , the maximum window is set to 64), the distance values are multiples of two. Thus, these two-search patterns use 8 points at each loop. The distance of the minimum distortion point is stored in the "Bestdistance" variable. The patterns used are either square or diamond depending on the configuration file.

978-1-5090-1645-7/16/$31.00 ©2016 IEEE



(a).Diamond search pattern

(b). Square search pattern

(c). Raster search pattern with iRaster equals to 4

Figure 2. Diamond/Square/Raster search pattern

C. Raster Search

When the previous search is completed, the next step is the raster search, which chooses the best distance corresponding to the best-matched point from the last search. Depending on this distance, three different cases are presented. First, if the best distance is equal to zero, then the process is stopped. Second if 1<Bestdistance<iRaster, a refinement stage is proceeded directly. In fact, the "iRaster" is a parameter that represents the central point and the first resulting from the research that it should not exceed. Third, if Bestdistance>iRaster which is set appropriately, a raster scan is achieved using the iRaster's value as the stride length. The raster search is performed when the difference obtained between the motion vectors from the first phase and the starting position is too large. This raster search is carried out on the complete search window. A FullSearch algorithm is shown in Figure 2 (c) for a 16x16 search window with iRaster equal to 4.

D. Star/Raster Refinement

When the motion vectors distance obtained in the previous steps is not equal to zero, the refinement step is performed. There are two types of refinements:

* Star refinement: The starting point of this search is the best point resulting from the previous searches. This step can also be done using either diamond or square pattern with different distances from one to "searchrange". This is done by multiplying the distance by two in each iteration. Hence, at a distance equal to one, the search for two adjacent points is applied in order to choose the best estimated motion vector having the best Sum of Absolute Difference (SAD).

* Raster refinement: Concerning this type of refinement, the starting point of this search is the best point resulting from the previous searches. This can be done using either diamond or square pattern by dividing the distance by two in each iteration up to having a distance equal to one. Where, the search for two adjacent points is applied and the process is stopped.

III. PROPOSED FAST CONFIGURATION

Many fast mode decisions were used in this work to reduce computational complexity of the ME algorithm using diamond search pattern, which reduces the computational complexity of all the encoder. Early CU termination (ECU), Early Skip Detection (ESD) and Coded Block Flag algorithm (CBF) are the methods used. An overview of the different configuration options adopted in HEVC test model HM16.2 is presented:

ECU: Enables the use of early CU determination, skipped CUs will not be split further [10].

ESD: Enables the use of CBF-based fast encoder mode. Once a 2Nx2N CU has been evaluated, if the RootCbf is 0, further PU splits will not be evaluated. Check inter 2Nx2N first and terminate if the motion vector difference is equal to zero and coded block flag equal to zero [11].

CFM: Enables the use of early skip detection and the skip mode will be tested before any other mode. Terminate encoding decision if the PU partitioning has a coded block flag equal to zero [12].

All of these additional modes influence only the inter picture prediction mode decision.

IV. EXPERIMENTAL RESULTS

A. Experimental conditions

In order to evaluate the performance of the fast configuration, the recent available HEVC reference software (HM 16.2) was implemented with the configuration Random Access Main profile (RA), with all previously described ECU, ESD, and CBF methods. The proposed implementation was compared to the original algorithm in terms of PSNR, bit-rate and encoding time speedup. The maximum search range used in this work is 64 with maximum CU Size equal to 64 and maximum CU partition depth set to 4. The simulations were carried on windows 8 OS platform with Intel ®core TM i7-3770 @ 3,4 GHz CPU and 12 GB RAM. For the different classes A, B, C and D with four resolutions [13] (2560x1600 /1920x1080/ 832x480 and 416x240 formats), the number of frames is set to

50. The fast algorithm was evaluated for quantization parameter (QP) 22 and 37.

B. *Evaluation criteria*

The result of the fast configuration used in this work is shown in Table I.

TABLE I. ENCODING TIME COMPARISON RESULTS BETWEEN FAST AND ORIGINAL HEVC CONFIGURATION

Class	**Sequences**	**QP**	**ΔEncoding Time (sec) ΔT(%)**
Class A 2560x1600	PeopleOnStreet	**22**	25
		37	49
Average			**37**
Class B 1920x1080	BQTerrace	**22**	29
		37	84.53
Average			**56.75**
Class C 832x480	RaceHorses	**22**	11.46
		37	44.17
Average			**27.81**
Class D 416x240	BQSquare	**22**	34.64
		37	75.26
Average			**52.25**

The data analysis of encoding time comparison is calculated using the following formula:

$$\Delta T(\%) = \frac{T_{Original} - T_{Fast}}{T_{Original}} \times 100\ (\%) \tag{1}$$

As consequence, the fast HEVC configuration allows saving up to 56.75% of encoding time obtained exactly for the BQTerrace 1920x1080 sequence. Further, for all test sequence used, the speedup is more important for higher QPs due to the fact that skip mode is more likely to be chosen for important values of QP [14]. The improvement is more important for videos containing low motion activities contrarily for high activity sequences such as PeopleOnStreet and RaceHorses.

Table II shows the PSNR (dB), ΔPSNR (dB), bit-rate (kbps) and the Δbit-rate(%) values obtained for original HEVC configuration compared to the fast one, used in ME for QP 22 and 37. ΔPSNR and Δbit-rate are calculated using the both following formulas:

$$\Delta PSNR = \mathrm{PSNR}_{Original} - \mathrm{PSNR}_{Fast}\ (\mathrm{dB}) \tag{2}$$

$$\Delta bitrate(\%) = \frac{bitrate_{Original} - bitrate_{Fast}}{bitrate_{Original}} \times 100(\%) \tag{3}$$

The results show a slice degradation in video quality in terms of PSNR, around 0.083 dB, with a little decrease in the bit-rate, around 1.25%.

Which proves that the adopted configuration is much efficient that the original, since it offered an important time saving, in addition to the decrease in bit-rate.

TABLE II. CODING EFFICIENCY WITH THE FAST CONFIGURATION FOR 50 FRAMES

Sequences	**QP**	**PSNR(dB)**			**bit-rate(kbps)**		
		Original_Conf	Fast_Conf	**ΔPSNR**	Original_Conf	Fast_Conf	**Δbit-rate(%)**
PeopleOnStreet 2560x1600	**22**	40.217	40.167	-0.04	33017.362	32859.76	-0.47
	37	31.21	31.173	-0.03	4738.104	4634.712	-2.182
Average				**-0.035**			**-1.326**
BQTerrace 1920x1080	**22**	37.99	37.938	-0.06	41970.95	41471.98	-1.18
	37	31.482	31.414	-0.067	1823.357	1785.09	-2
Average				**-0.063**			**-1.59**
RaceHorses 832x480	**22**	38.47	38.44	-0.029	5646.605	5621.98	-1.43
	37	29.387	29.155	-0.23	570.238	558.058	-2.138
Average				**-0.129**			**-1.28**
BQSquare 416x240	**22**	38.81	38.735	-0.075	1783.018	1767.44	-0.87
	37	29.25	29.112	-0.137	236.846	234.912	-0.81
Average				**-0.106**			**-0.84**

V. Conclusion

The new HEVC standard provides a significant improvement in video quality, especially for high-resolution videos. ME is the most challenging and consuming stage that causes a strong increase in computational complexity of all the encoder. Due to its new quadtree-based coding unit (CU) block partitioning structure, variable block size and multiple reference frames. In this paper, a fast configuration based on optimizing the motion estimation process using a diamond search and a fast mode decision algorithm for CU partitioning have been presented to reduce the computational complexity of the encoder. The simulation results prove that the fast HEVC encoder configuration allows saving up to 56.75% of encoding time obtained with a negligible degradation in video quality and decreasing the bit-rate.

The objective is to additionally reduce the motion estimation time with this fast configuration and with the parallelization of the search mode used here.

References

[1] ITU-T Recommendation H.265 and ISO/IEC 23008-2, ITU-T and ISO/IEC JTC 1, High Efficient Video Coding (HEVC), Apr. 2013.

[2] G. J. Sullivan, J. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp. 1649–1668, 2012.

[3] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards–including high efficiency video coding (HEVC)," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 22, no. 12, pp.1669-1684, 2012.

[4] K. Jongho, J. DongSan, J. Seyoon, C. Sukhee, S. C. Jin, K. Jinwoong, and A. Chieteuk, "An SAD-Based Selective Bi-prediction Method for Fast Motion Estimation in High Efficiency Video Coding," ETRI Journal, vol. 34, no 5, October 2012.

[5] N.Purnachand, L.N. Alves,and A. Navarro,"Improvements to TZSearch motion estimation algorithm For multiview video coding.," International Conference on Systems, Signals and Image Processing (IWSSIP), Austria, pp.388-301;2012.

[6] S.Liquan, L.Zhi, Z. Xinpeng, Z. Wenqiang, and Z. Zhaoyang, "Correspondence:an effective CU size decision method for HEVC encoders," IEEE Trans. Multimed. vol. 15, no 2, pp. 465–470, 2013)

[7] Software reference test model HM16.2 available in https://hevc.hhi.fraunhofer.de/, March 2016

[8] F. Belghith, H. Kibeya, H. Loukil, M.A. Ben Ayed, and N. Masmoudi,"A new fast motion estimation algorithm using fast mode decision for high-efficiency video coding standard", Journal of Real-Time Image Processing, vol. 11, no. 4, pp. 675-691, 2014.

[9] H. Kibeya, F. Belghith, H. Loukil, M.A. Ben Ayed,and N. Masmoudi, "TZSearch Pattern Search Improvement for HEVC Motion Estimation Modules", 1st International Conference on Advanced Technologiesfor Signal and Image Processing, 2014.

[10] K. Choi, S.-H. Park, and E. S. Jang, "Coding Tree Pruning Based CU Early Termination," document JCTVC-F092, JCT-VC, Torino, Italy, Jul. 2011.

[11] J. Yang, J. Kim, K. Won, H. Lee, and B. Jeon, "Early SKIP Detection for HEVC, document JCTVC-G543, JCT-VC, Geneva, Switzerland, Nov. 2011.

[12] F. Bossen, D.Flynn, K. Sharman, and K. Suhring, "HM Software," Manual October, 2014.

[13] A.Dıaz-Honrubia, J. Martınez, and P.Cuenca, "HEVC a review, trends and challenges," 2nd Workshop on multimedia data coding and transmission, 2011.