https://doi.org/10.2352/ISSN.2470-1173.2017.2.VIPC-400

Diamond Frequency Domain Inter Frame Motion Estimation for HEVC

Abdelrahman Abdelazim[A], Ahmed M. Hamza[B], Mohamed Hefeida[A], and Djamel Ait-Boudaoud[B]
American University of the Middle East, Kuwait City, Kuwait [A]
University of Portsmouth, Portsmouth, UK [B]

Abstract
This paper presents an efficient algorithm for motion estimation to reduce High Efficiency Video Coding (HEVC) standard encoding complexity. Phase correlation is initially utilized as a pre-processing step to indicate an approximation of the shift between coding units in the current frame and the reference frame. This is followed by a 9-point diamond search centered on the shift found in the initial step, in order to refine the best matching block. The proposed method has the potential to yield substantial improvements in terms of execution time and resulting video quality in comparison to the traditional search methods.

***Keywords** - Motion estimation; Phase Correlation; Diamond Search; HEVC Improvements; Motion Vectors; HEVC*

Introduction
The popularity of High Definition (HD) Video (and beyond) has been growing rapidly along with video streaming across networks with the associated enhanced capability of video capturing devices. This has fueled the development of the recent video compression standard HEVC (H.265) as a continuation of previously developed standards such as H.263 and H.264 [1]. Personal computers, tablets and mobile devices need to receive and display HD videos, and this became a major challenge for today's networks, particularly in terms of capacity, speed and throughput. In addition to addressing the above mentioned technology advancements, HEVC was designed to be backward compatible with H.264/AVC [2]. HEVC reduced the encoded video stream bit rate by approximately 50% while supporting higher resolution [3]. Quad Tree structure is used by HEVC standards as shown in Figure 1.

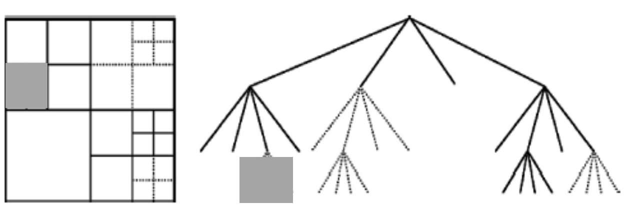

Figure 1: *CTU dividing into sub CUs [1]*

Each frame is divided into Coding Tree Units (CTUs) that includes a Luma Coding Tree Block (CTB) and two CTBs of Chroma. The selection of the CTU size is carried out in a predefined processing order. The resolution of the CTB varies to allow a trade-off between the quality of the encoded video and bit rate. Typical CTB sizes are 64×64, 32×32, and 16×16. Therefore, rather than macroblocks, the pictures of HEVC are distributed into coding tree blocks, as shown in Figure 2 in raster order. Each CTB is divided into tree structures, all the way down to 8×8. Coding Units (CUs) are considered the main unit used for prediction in HEVC and their size can be as follows: 64×64, 32×32, 16×16, and 8× 8.

An example of a 64×64 CU structure is shown in Figure 2.

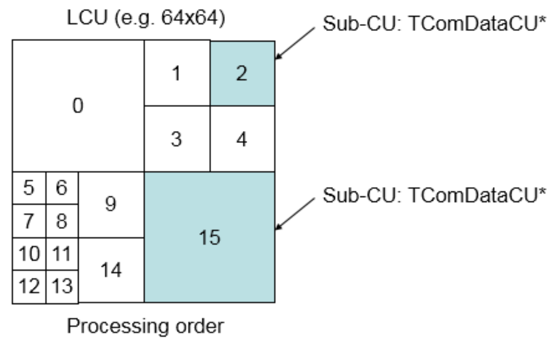

Figure 2: *CTU processing order of CUs*

The motion estimation complexity aspects and features were considered in HEVC standardization [4], nevertheless, the added sophisticated encoding tools remain a challenge when realizing HEVC in real-time applications. One of the main methods of reducing the redundancy between frames is inter-picture prediction [1]. This process involves selecting motion data by the encoder and utilizing it by the decoder to reconstruct the original video.

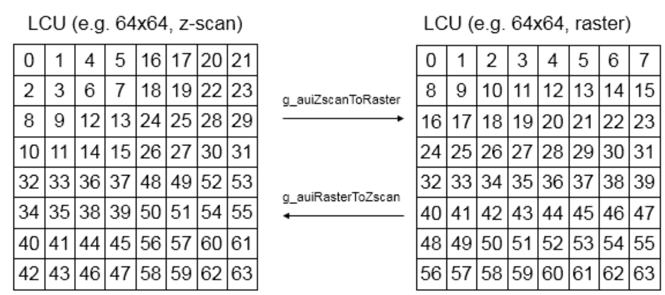

Figure 3: *Z-Scan order vs. Raster Scan Order HM15.*

Inter-Prediction requires heavy processing, and is considered the main factor in increasing video coding complexity [5]. Within inter-prediction, the motion estimation process is the main contributor to this increased complexity which accounts for more than 50% of coding time [6]. Despite this drawback, block-based motion estimation has been utilized in all previous standards of video coding.

This paper proposes an efficient algorithm contributing to the ongoing efforts to reduce video coding complexity in HEVC. The remainder of this paper is organized as follows: the following section provides a literature review of recent work in the area, followed by a detailed description of the proposed method. The results of experimentation performed to verify the proposed algorithm are presented thereafter. Finally, the last section concludes the paper and discusses possible improvements.

Literature Review

Temporal redundancy is effectively removed by block-based motion estimation; therefore, it is widely used in video coding. The process starts by performing block matching at the encoder level to find similarity between frames, best found matches is indicated by motion vectors. Block matching can be divided into three levels, namely: full, half, and quarter pel. The accuracy samples of half-pel and quarter-pel are generated by interpolation filtering, which is performed for all reference frames. Integer pel is obtained at the first step of most known algorithms, followed by sub-pel (half-pel and quarter-pel), which is limited to positions around the integer pixel's best match.

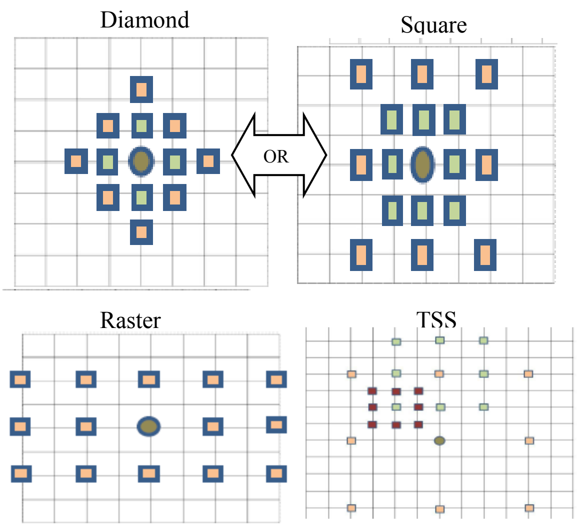

Figure 4: *Fast Search Methodology principles, Diamond or Square, in three step search, mixed, tuned and refined by final raster search*

The Sum of Absolute Differences (SAD) is the main distortion metric utilized for integer-pel motion search. The Sum of Absolute Transformed Differences (SATD), which is more computationally demanding, is utilized for half-pel and quarter-pel motion search. This is the default encoder setting in the HEVC reference software configuration file.

While exhaustive motion search has been considered as the most accurate one for a number of years, many new efficient algorithms have been tested and implemented in the standard reference software. Most of them use the same principles with different search patterns. An example of an efficient search algorithm is illustrated in figure 4. The method compromises of a three step to decrease the search point at the integer-pel motion search level [7]. The starting search position is determined by the Predicted Motion Vector (PMV) obtained from previously encoded neighboring CUs. As indicated in the figure, there are three search patterns which are can be used based on the required accuracy. The algorithm can be summarized as follows:

Step 1: Eight Blocks around the center area are selected for comparison.

Step 2: In this step, the size is reduced (usually by 50%). The center is moved to the point with the minimum distortion.

Steps 1 and 2 are repeated till the step size becomes smaller than one [8]. Different search schemes have been adopted, Diamond, Square, Hexagon, Cross, Rotating Shapes, to reduce the number of tested blocks between current and reference frames and provide fast coverage for most parts of the search region [9]. Test Zone Search (TZS) is the fast ME algorithm implemented in the HEVC reference software [10]. The algorithm consists of four stages. The first stage is the prediction followed by applying search patterns to find the global minimum point. Thresholds are used in the third stage to terminate the search process based on the results form step 2. In step 4, further refining criteria are added to ensure the resulted best match is the optimal or near optimal.

In this paper we take advantage of frequency domain motion estimation to extract the initial step. Frequency domain ME obtains accurate motion in a fraction of the time consumed by block matching. Then to refine the result and maintain the standard low bit rate, diamond search is applied to validate or to refine the results. The steps of the proposed platform are explained in details in the following sections.

Phase Correlation and Application in HEVC

The amount of image shifting done in the time domain can be best represented by measuring phase correlation [10]. Phase correlation will provide a simple estimation of rigid translational motion between two images. It has been demonstrated in numerous studies that the original method is the best method to identify integer pixel displacements, which has prompted the development of numerous sub-pixel displacement identification methods. Given two input images, the discrete 2D Fourier transform of both images is calculated as shown in Figure 5.

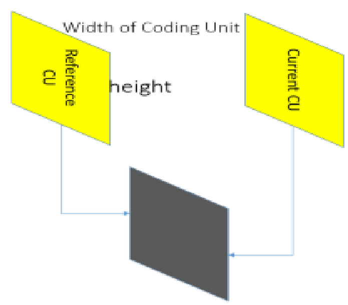

Figure 5: *applying phase correlation on CU related YUV buffers*

The basic principles are described below.

Assuming a translational shift between the two frames:

$$s_t(x,y) = s_{t+1}(x+\Delta x, y+\Delta y) \quad (1)$$

Their 2-D Fourier transforms are:

$$S_t(f_1,f_2) = S_{t+1}(f_1,f_2)\exp[2j\pi(f_1\Delta x + f_2\Delta y)] \quad (2)$$

Therefore the shift in the spatial-domain is reflected as a phase change in the spectral domain. The cross-correlation between the two frames is:

$$C_{t,t+1}(f_1,f_2) = S_{t+1}(f_1,.f_2)\cdot S_t(f_1,f_2) \quad (3)$$

The normalized cross-power spectrum is:

$$R_{t,t+1}(f_1,f_2) = \frac{S_{t+1}(f_1,f_2)\cdot S_t^*(f_1,f_2)}{|S_{t+1}(f_1,f_2)\cdot S_t^*(f_1,f_2)|} \quad (4)$$

From equation 2 and 4, we have:

$$R_{t,t+1}(f_1,f_2) = \exp[-2j\pi(f_1\Delta x + f_2\Delta y)] \quad (5)$$

The 2-D inverse transform is given by:

$$c_{t,t+1}(x_1, y_1) = \delta(x_1 - \Delta x, y_1 - \Delta y) \quad (6)$$

The displacement can be found by using the location of the pulse in equation 6. The maximum correlation, achieved when the two images are identical, has a value of 1 at (0, 0) [11].

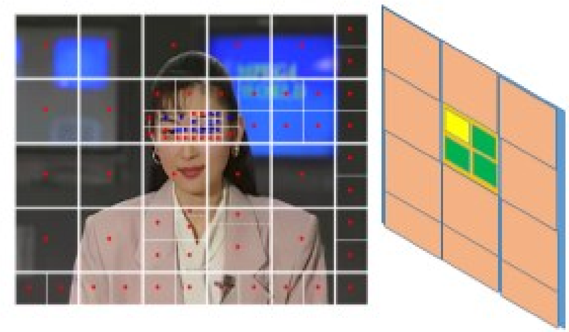

Figure 6. *CU Mapped to YUV Picture from Akyio*

In our implementation, the FFTW library with OpenCV were used for image processing, then applying the phase correlation steps mentioned above, to finally extract the shift between CUs. The Phase Correlation is applied to part of the frame and the reference frame, not the entire image. This is due to the fact that special change is not always uniform in one direction. In most motion related frames, different parts of the image move in different directions, thus making phase correlation results not very useful for inter-frame shift estimation, when applied on an entire frame. However, applying it to Coding Units Mapped Coordinates increases the probability that the shift in motion between current part of the frame and the reference actually yields the actual shift. This allows the use of the result as a starting point for a more dedicated search algorithm such as diamond search.
The general algorithm operation is highlighted in Algorithm 1.

Phase correlation methods can be applied very effectively from performance point of view. Several measures could be taken into account to improve the performance of such methods, where extensive application on every CU means that any tuning of the process could yield a substantial improvement in final time related measures.

Algorithm 1: General Operation of Proposed Method

1. *CF← Extract YUV from Current Frame*
2. *RF← Extract YUV from Reference Frame*
3. *YCF← Y-component buffer Pointer (CF)*
4. *YRF← Y-component buffer Pointer (RF)*
5. *Initialize FFTW and allocate memory for FFTW arrays*
6. *F-YCF← FFTW (YCF)*
7. *F-YRF← FFTW (YRF)*
8. *Obtain Cross Power Spectrum info. from previous frame*
9. *FCA ← Obtain Phase correlation Array*
10. *OpenCV ← FCA*
11. *Get Peak Pixel Location from 10*
12. *Deallocate Memory*
13. *Use step 11 as a starting location for motion vector search with the custom diamond (proposed method)*

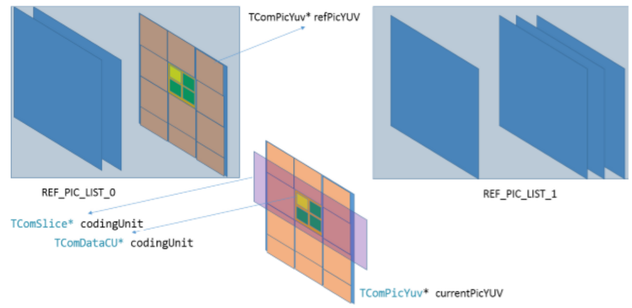

Figure 7. *Applying phase correlation on a part of YUV picture buffer*

Diamond Search over Phase Correlation

The previous section explained the method used to apply phase correlation over coding units maps over the YUV picture of current and reference frames. This results in one point representing the shift of the current part of the frame in relevance to the previous one. This point is then fed into a custom diamond search method to finalize the tuning process of finding the optimal point that yields the smallest SAD (Summation of Absolute Differences).

The applied Custom Diamond Search method uses a 9 point diamond search with a variable step size halved each iteration, and the best motion vector is placed at the center of the new diamond. The initial center of the diamond is chosen to be in the middle of the search region provided by the algorithm in a previous step, shifted by the results of the phase correlation.

Algorithm 2: Applying Diamond Search over Phase correlation

1. *Get peak x and y from phase correlation applied on CU Mapping YUV buffer data over Y component only*
2. *Get Center of search window*
3. *Add shift amount to center and make this point the starting search point of the diamond*
4. *Set distortion parameters*
5. *Set Max = maximum number of loops* ***(limit)*** *//to avoid deadlocks*
6. ***REPEAT*** *// For all 9 points of diamond*
7. *Calculate point location*
8. *Calculate SAD over point*
9. *Update Best_MV = MV with minimum SAD value*
10. *Best_MV == Center*
11. *Diamond step = Diamond step /2*
12. ***UNTIL*** *Diamond step < 1* ***OR*** *limit reached*
13. *Return MV with best SAD value*

Experimental Results

This section illustrates the experimental results obtained from applying the proposed method over several sequences of YUV videos. A comprehensive set of experiments has been carried out to validate the proposed algorithm. The testing environment was set on a PC running a 64-bit Windows 10 operating system. The system has and Intel(R) Core™ i5-4200U CPU @ 1.60GHz 2.30 GHz with an 8.00 GB RAM.

Table 1. Comprehensive test runs of proposed method against Full and Fast Search for various video sequences

Sequence	Full Search			Fast Search		
	BD-Bitrate (%)	*BD-PSNR (dB)*	*Time Saving (%)*	*BD-Bitrate (%)*	*BD-PSNR (dB)*	*Time Saving (%)*
Traffic 2560x1600	1.63	-0.2	35.08	1.01	-0.03	15.6
BQMall 832x480	2.25	-0.33	42.57	2.25	-0.12	18.04
Racehorses 832x480	1.89	-0.06	39.24	1.56	-0.08	16.59
BasketballPass 416x240	2.68	-0.14	46.29	1.97	-0.1	17.73
KristenAndSara 1280x720	1.07	-0.08	33.54	0.86	-0.05	12.84
ChinaSpeed 1204x768	1.49	-0.04	40.56	1.08	-0.06	14.18
Average	**1.84**	**-0.14**	**39.55**	**1.46**	**-0.07**	**15.83**

Six standard test sequences representing different categories of video have been selected from the Joint Collaborative Team on Video Coding (JCT-VC) test set. Table 1 shows detailed comparison by sequence. Time savings of the proposed method against the non-optimized Full Search are expectedly larger than against FastSearch, reaching above 40% in some sequences, although the bitrate/compression penalty is somewhat similar, the margin being within 0.5% on average, between the two.

The preliminary comparison results between the standard full search method and the proposed algorithm, are summarized in Table 3. Comparison criteria are Bit-rate, Quality (as PSNR), and Execution Time.

Table 2: Summary of comparisons between proposed techniques to standard search mechanisms in HM software

Criteria	*Fast Search*	*Full Search*
Execution Time (%)	-10.3%	-41.5%
Bit Rate (%)	+1.8%	+2.2%
PSNR	-0.09%	-0.12%

The negative values represent the Execution Time and PSNR degradation, the positive values represent the increment in the Bit Rate. It can be seen that our proposed algorithm achieves an average of 41.5% time saving with a negligible PSNR losse around 0.12 % and slight increments in the bitrate of roughly 2.2 %, compared to the standard Full search algorithm. Against the built-in Fast Search (TZSearch method) the proposed algorithm has a similar performance.

Conclusion

The main objective of this work was to optimize HEVC video encoding speed, at the same quality with a noticeable improvement in execution time. The Diamond Phase Correlation Search presented has shown good performance as an incremental solution to the optimization problem, working in line with other standalone techniques in the motion estimation procedure of HEVC. A variety of approaches is needed for a real-time speed encoding of high resolution video in modern applications. HEVC is the major current video coding standard with various characteristics that make it suitable for the task, hence our adoption of the standard implementation as a base for optimization. Results show little effect on the quality of the encoded video, and a similarly small (~2%) penalty in bitrate as a result of the speed gains presented – roughly 90% reduction in time over the Full Search base method.

References

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

[2] S. Gary J, O. Jens-Rainer, H. Woo-Jin, and W. Thomas , IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 12, DECEMBER 2012

[3] P. Gutenberg, "High Efficiency Video Coding | Project Gutenberg Self-Publishing - eBooks | Read eBooks online", Gutenberg.us,2016.[Online].Available: http://gutenberg.us/articles/high_efficiency_video_coding. [Accessed: 22- May- 2016].

[4] B. Frank,B. BenjaminS. Karsten, F. David HEVC Complexity and Implementation Analysis, PRE-PUBLICATION DRAFT, TO APPEAR IN IEEE TRANS. ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, DEC. 2012

[5] J. Kim, D. S. Jun, S. Jeong, S. Cho, J. S. Choi, J. Kim, C. Ahn, "An SAD based selective bi-prediction method for fast motion estimation in high efficiency video coding," in ETRI Journal, vol. 34, no. 5, pp. 753-758, Oct.2012.

[6] L.-M. Po, K.-H. Ng, K.-W. Cheung, K.-M. Wong, Y. M. S. Uddin, and C.-W. Ting, "Novel Directional Gradient Descent Searches for Fast Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 8, pp. 1189–1195, 200

[7] Efficiency and V. Coding, "High Efficiency Video Coding," 2009

[8] Davis P, M. Sangeetha Implementation of Motion Estimation Algorithm for H.265/HEVC, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 3, Special Issue 3, April 2014

[9] H. Jia and L. Zhang, "Directional Cross Diamond Search Algorithm for Fast Block Motion Estimation," arXiv Prepr. arXiv0806.0689, pp. 1–23, 2008.

[10] N. Purnachand, L. N. Alves, A. Navarro, "Fast motion estimation algorithm for HEVC," IEEE, ICCE,Berlin, 2012. [9] Z. Pan, Y. Zhang, S. Kwong, X. Wang, L.Xu, "Early termination forTZSearch in HEVC motion estimation," Int. Conf. ASSP 2013, pp 1389-1393. [10] H. S. kim, J. H. Lee,

[11] 7. J. J. Pearson, D. C. Hines, S. Goldman, and C. D. Kuglin, "Video rate image correlation processor," Proc. SPIE 119, 197–205 (1977).

[12] A. Abdelazim, S. J. Mein, M. R. Varley, C. Grecos, and D. Ait-Boudaoud, "Phase correlation based adaptive mode decision for the H.264/AVC," Processing, vol. 7871, p. 78710O–78710O–6, 2011.

Author Biography

Abdelrahman Abdelazim *is an associate professor of Computer Engineering at the American University of the Middle East (AUM) in Kuwait. He holds a BEng (Hons) degree in Digital Communication and a PhD degree in Engineering, both from the University of Central Lancashire (UCLAN), Preston, UK. Between 2008 and 2012 he worked as Lecturer in Electronics within the School of Computing, Engineering and Physical Sciences (CEPS). His experience includes leading the implementation of a number of academic and industrial digital communication projects. His research interests are in the area of reducing the complexity of Video Coding Encoders in real-time Scalable and Multi-view applications and the area of teaching and learning in higher education. Abdelrahman is member of the IET since 2006. He is a chartered engineer and a Fellow of the Higher Education Academy.*

Ahmed Hamza *is a PhD student in the School of Computing at Portsmouth University, where he is pursuing research in real-time video encoding optimization by deep learning networks. He obtained a M.S in Computer Science from Georgetown University in 2010, with a thesis on structural similarity algorithms in chem-informatics. His interests are in Video Coding, algorithms, information theory, and natural systems. Ahmed is a member of IET and IEEE.*

Mohamed Hefeida *is an Assistant Professor and Chair of the Computer Engineering department at the American University of the Middle East (AUM) in Kuwait. He holds a PhD in Electrical and Computer Engineering from the University of Illinois at Chicago (2013), where he also worked as a visiting assistant professor before joining AUM in 2014. His research interests span a wide spectrum of networking and communications techniques, with particular interest in energy efficient communication and processing techniques for resource restricted environments.*

Djamel Ait-Boudaoud *joined the Portsmouth University in 2010 and is currently Dean of the Faculty of Technology. Before Portsmouth he was the head of the School of Computing, Engineering and Physical Sciences at the University of Central Lancashire for close to 10years. His research interests are predominantly focused on the problems of optimization with applications in 3D computer vision, video standards (H264) and solving combinatorial (ordered-sequence) problems using evolutionary algorithms. He gained a PhD in 1991 from the Department of Electrical and Electronic Engineering at University of Nottingham, UK., is a Chartered Engineer (CEng) and a fellow of the Institution of Engineering and Technology (FIET). He is currently a committee member of the Engineering Professors Council (EPC) and serve on the reader's panel for the Queens Anniversary Prize for Higher and Further Education, UK.*