

Cross Search Frequency Domain Motion Estimation Algorithm for the High Efficiency Video Coding Standard

Abdelrahman Abdelazim^A, Ahmed M. Hamza^B, Djamel Ait-Boudaoud^B

American University of the Middle East

Kuwait City, Kuwait ^A

University of Portsmouth, Portsmouth, UK ^B

Abstract— Development in video coding technology is at the heart of recent rapid advances in streamed and live digital video technologies. This paper proposes the implementation of the Large Cross Search Algorithm in the H.265/High Efficiency Video Coding motion estimation process, a technique common to most hybrid block-based video encoders. Our speed optimization through frequency domain phase-correlation enables the user to compress a video in a short time while maintaining high image quality with bit rates similar to the standard implementation. The implementation was tested with preliminary results on common video sequences.

Keywords— Video codec; Motion Estimation; Large Cross Search

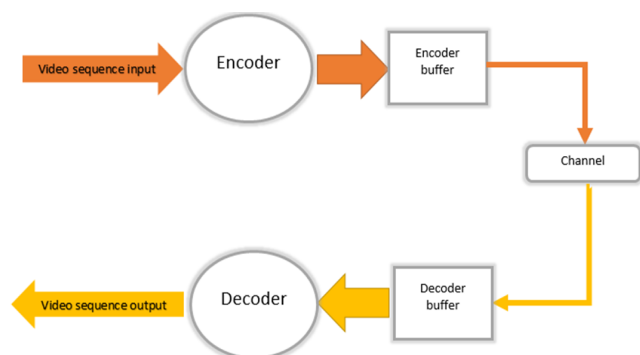


Figure1: Video Coding Process

I. INTRODUCTION

In recent years, video coding standards groups have built on prior methods accumulated through MPEG, H.263, and H.264, attempting to further eliminate some redundancy in data from encoded video sequences [2]. These standards had some limitations such as having small encoding blocks. In January 2013, the latest video coding standard for video compressing techniques was created, the H.265 codec [3,4]. H.265 is known as HEVC which stands for High Efficiency Video Coding.

HEVC has advanced features, improving on the compression efficiency when compared to previous standards, and supports larger encoding blocks which was the limit of previous standards. It can start with a 64×64 block, and if there are too

many differences, this block can be portioned into four 32×32 blocks and so on. As a result, it has more partitioning structures compared to prior codecs. Although it is the latest video coding standard, the inevitable downside here is its computation complexity. HEVC encoder flexibility stems from the fact that it contains an increased number of coding tools, beyond those provided by earlier video coding standards [5], and a large array of different modes of operation (slicing, block search type, search depth level, etc.) which are configurable at runtime.

II. OBJECTIVES

In this paper, we aim to analyze and improve the HEVC motion estimation process by the introduction of the Large Cross Search algorithm, which we experimentally validate on video sequences from the standard test set. A broader aim is to develop the current understanding of block-motion search optimization and how it affects the motion estimation process.

III. EXISTING SEARCH METHODOLOGIES

Several problems are encountered by videos which have large storage requirements in modern settings, resulting in a growth in the bit rate and the execution time to encode a video. A video codec is typically lossy [6] – that is, information is not maintained fully in its original form, or in equivalent derivatives of it. This leads to signal noise and loss of quality. One can attempt to solve these problems by using the Large Cross Search Algorithm. This algorithm maintains quality level of the video while resulting in significantly less time than the reference execution time for the encoding process.

Video encoding methods are applied to reduce redundancy in video data without adversely affecting video quality. Motion estimation module is at the heart of system for inter-frame coding. It is the process of determining the best matched block of the current frame in the search region of reference from temporally neighboring frames. Moreover, it helps to discover the relative motion between two images in order to eliminate temporal redundancy. In reality, motion estimation based

encoders are the most widely used in video compression techniques [7]. Several techniques of motion estimation have been created, such as the block matching technique and the phase correlation technique.

A. Block Matching Technique

Block Matching is a technique that works to find matching macroblocks in a frame of digital video frames. This technique works by partitioning the current frame of the video into macroblocks, then comparing each macroblock with the corresponding block in the reference frame of the video in order to get the first difference between these blocks. The process is going to be repeated for each row of the reference frame that is completed. The procedure sums all pixel differences to obtain the Summation of Absolute Differences (SAD). In this setting, the SAD is the distortion measure in the Rate Distortion cost function shown below, with the product of λ and Motion Vector (MV) bit-cost shown as

$$J = \text{SAD} + \lambda \cdot \text{MV} \quad (1)$$

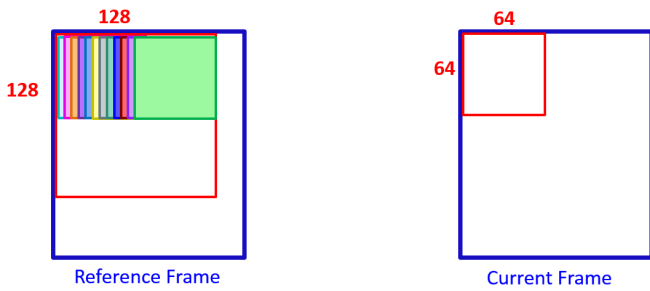


Figure 2: Cost Function

The same steps are repeated for different rows while taking the least value of J , where J is the cost function for the particular mode of prediction used. The process takes place in the block matching of different width blocks in a search area as shown in Figure 2.

B. Phase Correlation Technique

Phase correlation is the technique that is used to decide the motion vectors which are used in the motion estimation. It locates the integer pixel movement between two images and gets the shift point. It works by a shift in the spatial domain which is equivalent to the phase shift in the frequency domain, and the resulting output is a value that is in a range from zero to one depending on the similarities between the two images. There are three cases. The first case is if this value is equal to

zero. Then, there is no resemblance between the two images due to the huge change of scene between them. The second case is if the value is equal to one, then these two images are exactly the same due to the zero motion between them (complete redundancy of information). The third case is if the value is in the range of (0.1–0.9), then there are some similarities between the two images depending on the value. The higher the resulting value, the higher the similarities between the two images and the less motion between them. In addition, the way to calculate the Phase Correlation is based on mathematical analysis. Visual Studio compiler is the software of the phase correlation code that was compiled in order to calculate the phase correlation position and value.

• Mathematical Analysis

The algorithmic details behind the phase correlation technique are illustrated below. The phase correlation technique deals with two frames that have to be converted to a frequency domain to measure the movement between them. The analysis starts with presuming a translated shift between the two frames [9].

$$s_t(x, y) = s_{t+1}(x + \Delta x, y + \Delta y) \quad (2)$$

Then comes the Fourier Transform step:

$$S_t(f_1, f_2) = S_{t+1}(f_1, f_2) \exp[2 * j\pi(f_1 \Delta x + f_2 \Delta y)] \quad (3)$$

Then, we calculate the cross correlation between them:

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

The cross-power spectrum is computed, as follows:

$$R_{t,t+1}(f_1, f_2) = \frac{S_{t+1}(f_1, f_2) \cdot S_t^*(f_1, f_2)}{|S_{t+1}(f_1, f_2) \cdot S_t^*(f_1, f_2)|} \quad (5)$$

We substitute equations (3) and (5):

$$R_{t,t+1}(f_1, f_2) = \exp[-2j\pi(f_1 \Delta x + f_2 \Delta y)] \quad (6)$$

The final step applied is discrete Fourier Transform:

$$c_{t,t+1}(x_1, y_1) = \delta(x_1 - \Delta x, y_1 - \Delta y) \quad (7)$$

As shown in equation (7), the shift point was found.

IV. LITRETURE REVIEW

Over the past 20 years, many algorithms were put forth to reach a permissably low computational complexity while preserving quality with the same low bit rate as the various standards. We group them together to discuss common factors between them. The are three main factors that affect the output encoding of raw video: objective quality, bitrate, and complexity of the encoder. The proposals in the Video Coding Experts Group (VCEG) have highlighted different techniques of motion estimation, most of them geared towards having high efficiency leading to a high quality of significantly reduced bitrate counts at the same resolution and frame rate of encoded video. The overall complexity of the encoding process tends to rise as these were sucessively added to the standards. With speed becoming an important real-time encoding consideration, the full search could use up to 70% of the complete computation of the video compression process. [12].

Alternatives in the literature include the three-step search (low quality), the four-step search and the DGDS [13]. The four-step search has significant improvements in motion detection over the three-step search pattern [14]. The DGDS stands for Directional Gradient Descent Search. Under the low-quality parameter, there comes the three-step search, and under the medium-quality parameter, there comes the four-step search. Beneath them are the fixed-shape patterns such as cross search, diamond search and hexagon search. Below them are two methods to follow either integer pixel or sub-integer pixel.

For the sub-integer pixel, there exist granularity levels of half and quarter-integer pixel. The half-integer pixel is a more accurate level. A third step in motion search is the high quality geared, higher complexity combination of the fixed shape patterns such as MCDH (Modified Cross-Diamond-Hexagonal) and HEXFS [15] which stands for Hexagonal pattern.

V. LARGE CROSS SEARCH ALGORITHM

The Large Cross Search Algorithm is applicable to the motion estimation part of inter-mode coding in the High Efficiency Video Coding standard (H.265). It has to pass through some stages in order for it to be implemented as shown in Figure 3. The search method involves two frames: the reference and the current frames as shown in Figure 4. It is under the frequency domain because it depends on both the phase correlation technique and the cross-shape search. It assumes action when there is a shifting with a frame that changes its place to another. This algorithm starts with the phase correlation technique to find the offset of how much a frame moved from one place to another, and get the shift point. Moreover, this technique depends on calculations, which are Fourier Transform

calculations that were mentioned in the mathematical analysis. Phase correlation is featured with the least execution time when encoding the video, but with having low quality video. After finding this shift point, the center of the search area is going to be found next as shown in Figure 5. This is because this method has to have the center of the search region added to the point found through the phase correlation as shown in Figure 6. The next step is to place the large cross over this point which is the start search point in order to start searching for the best matched block as shown in Figure 7. The large cross consists of nine points including the center point. The minimum cost for each point will be calculated by the sum of absolute differences (SAD), and the point with the least cost will be the best match block. The cross shape is going to be repeated six times because whenever we increase the repetition in this method, it is noticed that the bit rate of the encoded video decreases.

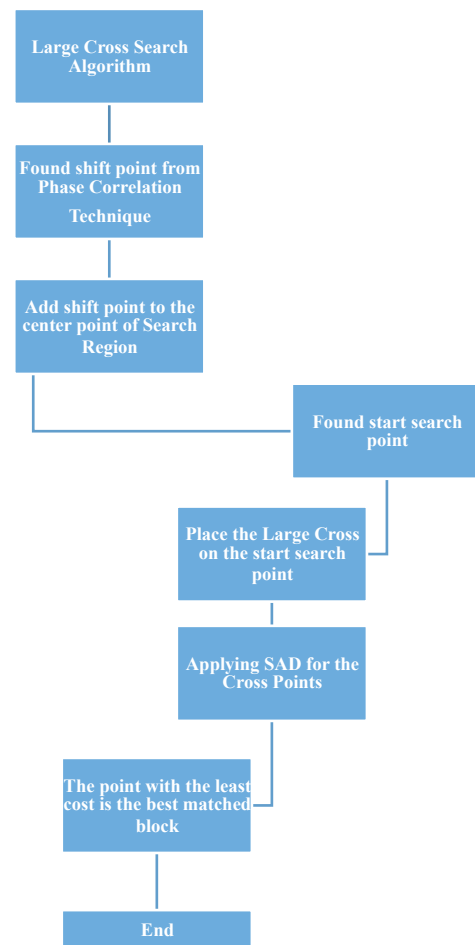


Figure 3. Flow chart of the Large Cross Search Algorithm



Figure 4: (a) Reference Image

(b) Current Image

V. EXPERIMENTAL RESULTS

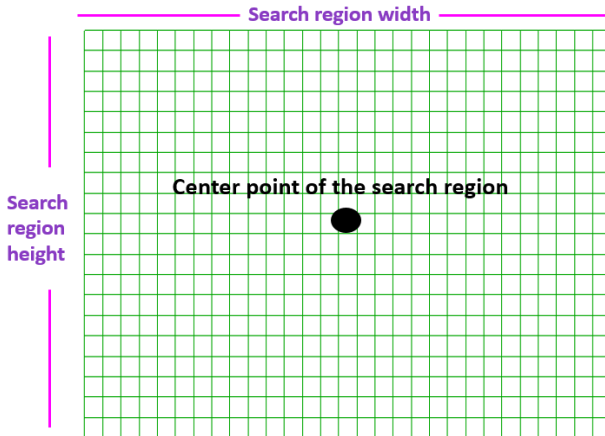


Figure 5: Locating the center point of the search region

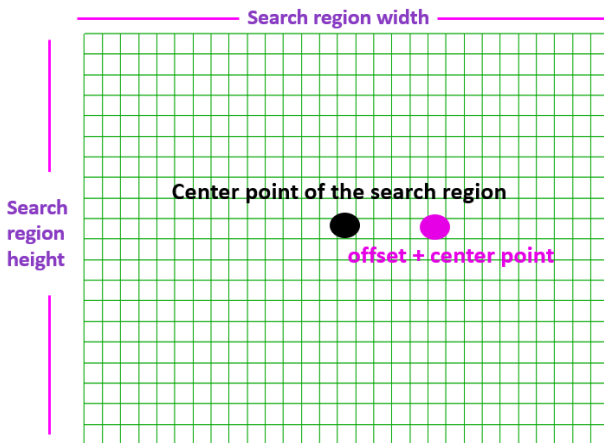


Figure 6: Adding the shift point to the center point of the search region

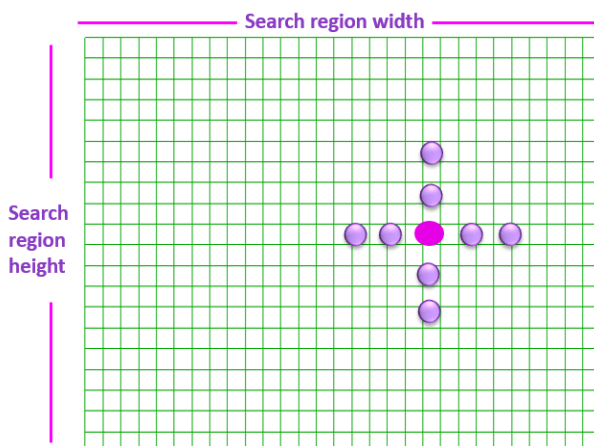


Figure 7: Placing the Large Cross on the start search point

The testing of the Large Cross Search Algorithm was done on a PC (Windows 10) with: Intel (R) Core (TM) i7-4700MQ CPU @ 2.40 GHz processor, main memory at 8GB RAM on a 64-bit operating system, and a standard magnetic hard-disk for video sequence storage.

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 is within 1.8% on average between the two. 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 38.2% time saving with a negligible PSNR loss around 0.18 % and slight increments in the bitrate of roughly 1.8 %, compared to the standard Full Search algorithm. Against the built-in Fast Search (TZSearch method), the proposed algorithm has a similar performance in terms of Bit Rate and PSNR, with average time saving of 13.55%.

I. 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 Large Cross Search algorithm 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.

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 40% reduction in time over the Full Search base method. We suggest further work in testing on a variety of configurations and video sequences, then combining this work with other optimization techniques in suggested additions to the standard.

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

Sequence	Full Search			Fast Search		
	<i>BD-Bitrate (%)</i>	<i>BD-PSNR (dB)</i>	<i>Time Saving (%)</i>	<i>BD-Bitrate (%)</i>	<i>BD-PSNR (dB)</i>	<i>Time Saving (%)</i>
Traffic 2560x1600	1.02	-0.03	42.82	2.23	-0.02	14.32
BOMall 832x480	1.59	-0.51	35.75	1.65	-0.13	16.78
Racehorses 832x480	2.22	-0.12	38.66	1.72	-0.15	12.31
BasketballPass 416x240	1.48	-0.04	43.12	1.09	-0.07	11.21
KristenAndSara	2.81	-0.29	36.56	2.46	-0.16	17.54
ChinaSpeed 1204x768	1.6	-0.07	32.33	0.89	-0.07	9.12
Average	1.79	-0.18	38.21	1.67	-0.1	13.55

REFERENCES

[1] 3GPP, "Video codec performance," no. 26.902, 2008.

[2] G. Held and T. R. Marshall, "Data Compression," 1987.

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

[4] 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.

[5] P. Nalluri, L. N. Alves, and A. Navarro, "Complexity reduction methods for fast motion estimation in HEVC," *Signal Process. Image Commun.*, vol. 39, pp. 280–292, 2015.

[6] 3GPP, "Video codec performance," no. 26.902, 2008.

[7] T. Aziz and D. R. J. Dolly, "Motion Estimation and Motion Compensated Video Compression Using DCT And DWT," vol. 2, no.12,pp.667–671,2012

[8] J. R. 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)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, 2012.

[9] 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. 787100–787100–6, 2011.

[10] T. Aziz and D. R. J. Dolly, "Motion Estimation and Motion Compensated Video Compression Using DCT And DWT," vol. 2, no. 12, pp. 667–671, 2012.

[11] I. Engineering, "Review and Comparative Study of Motion Estimation Techniques To Reduce Complexity in Video," pp. 3574–3584, 2013

[12] R. 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)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, 2012.

[13] 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, 2009.

[14] Lai-man Po and Wing-chung Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, 1996.

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