Modified Three Diamonds Motion Search for Video Coding

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

Abstract— the fundamental principle of Video Coding Standards is reducing the spatial and temporal redundancies from the consecutive frames through what is called motion Estimation. Efficient video coding standards have very sophisticated motion Estimation algorithms. However, those schemes are complicated and consume the majority of the encoding time. In this paper a new matching algorithm for fast block motion estimation is proposed which is a Modified Three Diamonds Search Algorithm using the Four Step-Search and the Half-Pel Diamond Search. It reduces the motion estimation time without affecting the video quality when compared to the existing algorithms. Experimental results demonstrate the efficiency of the proposed scheme.

Keywords— spatial redundancies; temporal redundancies; Motion Estimation; video coding; Modified Three Diamonds Search Algorithm; Four Step-Search; Half-Pel Diamond.

I. Introduction

Motion estimation (ME) is the process by which the relative motion of the block-based partitions in a video frame is estimated by comparisons with other parts of the picture, prior frames, future frames, and a combination thereof. It starts by locating objects within an image; then it computes their motion from one frame to another; then it represents these motions by motion vectors. The significance of this is eliminating the need to encode and transmit all the frame data in each respective picture. Instead only the motion vector is transmitted, which is the direction of the movement either "right, left, top or bottom" between the previous frame to the current frame of a video sequence along with the residual differences between the frames, which is reflected by the mathematical values between the pixel in the consecutive frame. Despite the fact that this is considered as a very efficient technique for the video compression, 80% of the encoding is consumed by the ME [1]. Furthermore, this is delaying the realization of video coding standards in real time applications. Therefore, reducing the computational complexity of the encoding process by efficient algorithms is required. Block matching using ME is implemented in all previous and current video coding standards: H.261, H.263, H.264 and H.265 [2]. The H.265, the High Efficiency Video Coding (HEVC) standard, is the latest video coding standard by Video Coding Expert Group (VCEG) [3]. It is the platform used in this paper. This work presents a modified algorithm that tackles the ME constraints. The proposed algorithm is divided into three stages; each is discussed in details in the following subsections.

Over the past 20 years, several algorithms were suggested to reduce the complexity of video coding standards, all of which are aimed at accomplishing this while maintaining the same quality and bit rate as the standards. Surveying the literature one can conclude that the large numbers of motion estimation methods have not reached an ideal solution; many of them manage to reduce the execution time but at the cost of reducing the quality of the compressed video and/or increasing the bit rate of the output video stream. Many use different shapes rather than the square (commonly used in video coding standards) such as a diamond, cross or hexagon as shown in figure 1 (a-b-c) respectively [4]; others use a modification or a combination of the previous methods such as Cross Diamond search or Cross Hexagon Diamond Search [5] as shown in figure 1 (d). A number of them are based on mathematical models [6].

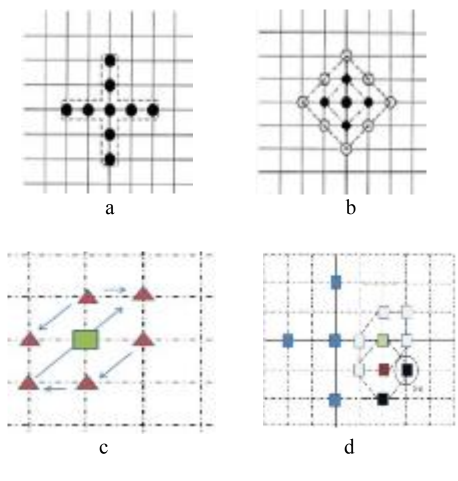

Figure 1 (a-b-c-d): Alternative proposed methods

This paper is divided as follows. Section 2 consists of the theoretical background which highlights the need of the video coding standard, providing an example of that. Section 3 explains the technique of the block matching process. Section 4 explains the implementation of the proposed method. Section 5 illustrates the experimental results of the proposed method while comparing it with the standard solutions. Section 6 concludes the paper.

II. THEORETICAL BACKGROUND

Processing and transmitting raw video sequences consumes bandwidth, which requires storage as well [7]. Table 1 provides an example of a raw video without any compression. The selected video has multiple resolutions compared with the required number of bits. If it is a one-second video with a resolution 640 × 480, it consumes 27 MB out of the hard drive storage [8]. According to this, the need of video coding standards becomes apparent [9].

Table 1: example of a raw video without any compression

Time / Size	640×480	320×240	160×120
1 sec	27 MB	6.75 MB	1.68 MB
1 min	1.6 GB	400 MB	100 MB
1 hour	97 GB	24 GB	6 GB
1000 hours	97 TB	24 TB	6 TB

III. BLOCK MATCHING

The ME algorithm consists of a prediction process and a block matching search process (BM) [10]. The prediction process starts by finding the initial search point which is assumed to be in the middle of the neighboring reference frame. Then, it uses the block matching process to locate the best matching block by using fixed search patterns which are considered as multiple squares. In a BM approach, the video sequence consists of image frames that are divided into blocks [10]. The aim of the BM is finding the best matching block of the current target frame inside the search region of the translated frame, while minimizing the matching criteria which is the Sum of Absolute Difference (SAD) as shown in Fig. 2. The problem with that is the time consumption, because finding the best match for a block in the current frame requires searching in the search region of the reference frame; and the search will be square that is been moved pixel by pixel until the search region is completed, after that dividing the block in the current frame even further. Moreover, instead of moving pixel by pixel, the pixel fraction is considered. This highlights the high processing time consumed by this process. A solution of this problem is the MTDS4SSHpelD that is proposed in the following section.

The process of the rate distortion optimization consists of several calculations of the distortion between the current and the translated sample values. Discovering the optimum coding parameters is performed in a rate distortion (RD). An RD optimization problem in this context can be expressed as: "$J = D_{SAD} + \lambda . R(MV)$" which is identified as the cost function. λ is the Lagrange multiplier. While MV is the motion vector representing direction of the block displacement, and the resulting rate R from encoding this information.

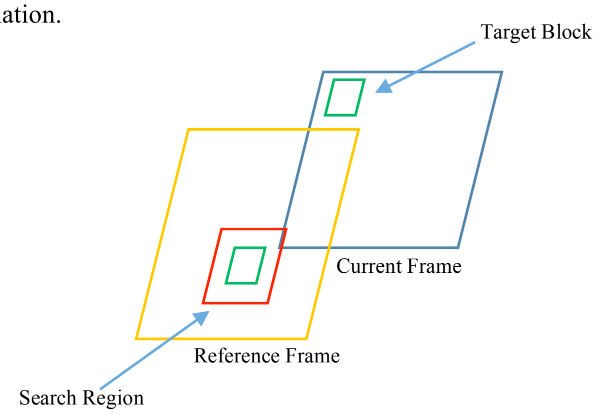

Figure 2: Block matching process

IV. PROPOSED ALGORITHM

The proposed algorithm is based on the modified diamond search and the 3-step search [8]. It is divided into three steps: firstly, a 4 Step Search Algorithm (4SS) is carried out, followed by a Modified Three Diamond Search Algorithm (MTDS); then a final stage consisting of sub pixel diamond search is implemented as shown in Figure 3.

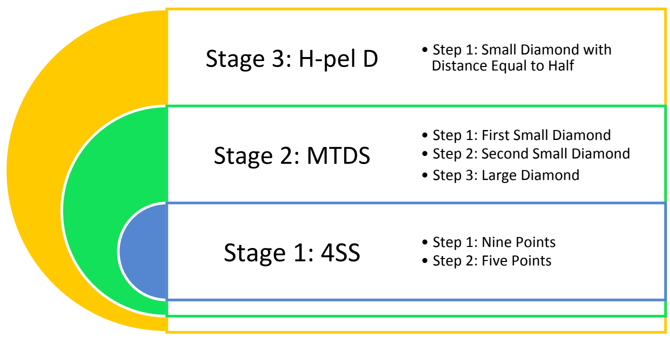

Figure 3: The Process of the MTDS4SS Diamond

- ***Section 1 4SS: Four Step Search***

The 4SS has two main parts centered around the predication point which is obtained from the neighboring blocks behavior. A square shape that includes nine points is checked for a best match as shown in Figure 4(a). Following this, 5 points centered around the best match are searched to find a better match as shown in Figure 4(b). This process is repeated four times to refine the optimal global best match. The matching criteria used is the Sum of Absolute Difference (SAD). The four-step search is proven to be more efficient than the proceeding three steps search as indicated in [13]. Following the 4SS a MTDS is carried out as explained in the next section.

- ***Section 2 MTDS: Modified Three Diamonds Search***

The MTDS places a small diamond over the obtained best match and searches for an even better match as shown in Figure 4(c). Then the process is repeated using another small diamond shape as shown in Figure 4(d). The final step of this process is replacing a larger diamond around the best match to verify the result or to find an enhanced match as shown in Figure 4(e). The difference between the two small diamonds and the large diamond is represented by the distance. The small diamond has a distance which is of one pixel of the center point; the large diamond uses a distance of two center points. Otherwise, if the distance is equal to two, the large diamond will occur. The concluded best match is used to be the starting point of the final step of the algorithm which is the Sub-pixel diamond search.

- ***Section 3 H-Pel D: Half Pel Diamond Search***

Similar to the standard, sub-pixel motion search is considered after finding the full pixel best match. However instead of searching the nine interpolated positions around the best match, five position are searched using sub-pixel diamond search as indicated in Figure 4(f).

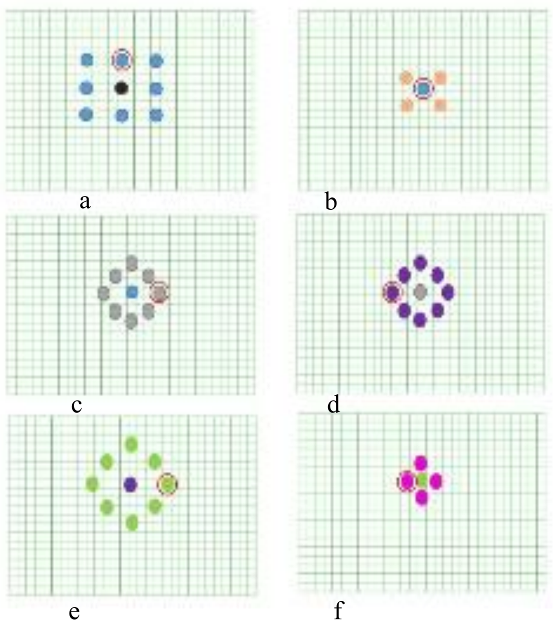

Figure 4
(a, b, c, d, e, f): The Implementation of the Proposed Method

V. EXPERIMENTAL RESULTS

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.

Three test sequences representing three different categories of video have been selected from the Joint Collaborative Team on Video Coding (JCT-VC) [1]. The JCT-VC classifies the test sequences into six classes A, B, C, D, E and F as shown in Table 2.

The selected sequences are: BaketballPass 416 Intel(R) 240, RaceHourses 832 Intel(R) 480, and BQMall 832×480. They were run using a created batch file which includes the combination of the command lines that represent the testing of the proposed methods against the standard solution. The average was taken for the three videos and then tested against the standard solutions through different comparison criteria which are bit rate (BR), PSNR, and execution time (ET). The reference software was compared against the proposed algorithm using the standard Bjorne-Delta settings for testing.

Table 2: The Test Sequences Classification

Class Sequence	Name Resolution
A	Traffic 2560x1600
B	BQMall 832x480
C	Racehorses 832x480
D	BasketballPass 416x240
E	KristenAndSara 1280x720
F	ChinaSpeed 1204x768

The experimental results are given in Table 3. In this table, the negative values represent the execution time and PSNR degradation. However, the positive values represent the increment in the bit rate. It can be seen that our proposed algorithm achieves an average of 92.6% time saving with a negligible PSNR losses around 0.19 % and slightly increments in the bit rate by 2.76 % compared with the standard Full Search algorithm.

Nevertheless, the proposed algorithm outperforms the Fast Search algorithm which is considered to be the fastest algorithm for the HEVC encoding process. The proposed algorithm has a saving in the execution time with an average of 4.63% with an increment in the bit rate by 2.46% and a slight decrement in the PSNR of approximately 0.17% which is considered to be negligible.

Table 3: MTDS4SS H pel D: Experimental Results

Sequence	Full			Fast		
	ET %	BR %	PSNR %	ET %	BR %	PSNR %
BQMall 832x480	-90	+2.5	-0.16	-3.7	+2.2	-0.15
Racehorses 832x480	-95	+3.3	0.25	-6.3	+3.1	0.23
Basketball Pass 416x240	-93	+2.5	0.17	-3.9	+2.1	0.14
Total (AVG)	-92.6	+2.76	-0.19	-4.63	+2.46	-0.17

VI. CONCLUSION

MTDS4SS Half-pel Diamond was presented in this paper, with a tested implementation and preliminary results. It considered an algorithm under the block matching set of techniques for HEVC coders. It consists of three main techniques: the 4SS, MTDS, and Half-pel Diamond, already in use in prior standards. A clear comparison was done according to three main criteria which are the execution time, bit rate and the PSNR. The proposed method was compared with the two standards solutions. At the end, the experimental results show that the MTDS4SS outperforms other algorithms in terms of reducing the execution time, while maintaining the bit rate and the PSNR.

VII. REFERENCES

[1] I. Jens-Rainer Ohm, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)," *IEEE transactions on circuits and systems for video technology,* vol. 22, p. 12, 2012.

[2] H. Efficiency and V. Coding, “High Efficiency Video Coding,” 2009.

[3] Bhavina Patel, "Review and comparative study of motion estimation techniques to reduce complexity in video compression," *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,* vol. 2, no. 8, 2013.

[4] R. Vani, Modified Cross Hexagon Diamond Search Algorithm for Fast Block Matching Motion Estimation. *International Journal of Engineering and Technology (IJET)*, vol. 6, 2014.

[5] A. Mohamed, "Subpixel Accuracy Analysis of Phase Correlation Shift Measurement Methods Applied to Satellite Imagery," *(IJACSA) International Journal of Advanced Computer Science and Applications,* vol. 3, p. 12, 2012.

[6] G. Held and T. R. Marshall, “Data Compression,” 1987.

[7] R.Vittal, "Estimation of Motion Vector Parameters using Modified Diamond Search," *International Journal of Computer Applications (0975 – 8887),* vol. 15, 2011.

[8] D. Salomon and G. Motta, “Video Compression,” Handb. Data Compression, pp. 855–952, 2010.

[9] Erik Cuevas1, "Block matching algorithm for motion estimation based on Artificial Bee Colony (ABC)," *Artificial Bee Colony (ABC), Applied Soft Computing Journal, pp. 3047-3059. ,* p. 12, 2013.

[10] P. Lathwal, "Line Searches for fast block motion estimation," *Proceedings of National Conference on Innovative Trends in Computer Science Engineering (ITCSE-2015) ,* 2015 .

[11] I. Aroh Barjatya, "Block Matching Algorithms For Motion Estimation," *DIP 6620 Spring 2004 Final Project Paper,* 2004 .

[12] D. Sathik, "Three step vs Four step Block matching search algorithm in Stereo image Compression," *International Journal of Advanced and Innovative Research,* vol. 1 , no. 2, 2012.

[13] Crowcoft, J., “How Big Is a Single Frame of Video?”, (1998). .

Figure 5: Algorithm Description for the Proposed Method