International Journal of Computing and Digital Systems
@ 2014: Scientific Publishing Center, University of Bahrain

A Fast Diamond Motion Estimation Search Algorithm for Real Time Video Applications

Yasser Ismail[1]

Computer Engineering Department, University of Bahrain, Bahrain

E-mail address: yasserali1977@gmail.com

Received 23 Feb. 2014, Revised 19 Mar. 2014, Accepted 27 Mar. 2014, Published 1 May. 2014

Abstract: This Motion Estimation (ME) has been widely used in most of the video standards. It is the most computationally intensive part of video compression. Speed up this process will open new real time applications. A novel Fast Diamond Search (FDS) algorithm is proposed in this journal. The proposed algorithm is a modified version of the conventional Diamond Search (DS) algorithm. Two main steps are added to the conventional DS algorithm for more computational savings achievement and keeping an acceptable coding efficiency (good bit-rate and Peak Signal to Noise Ratio of the transmitted video signal). The first step is to accurately remove the internal redundancy when calculating the Sum of Absolute Difference (SAD) between the current and the candidate blocks. This will be achieved using the Dynamic Internal Stop Search (DISS) algorithm.The second step aims to skip all the irrelevant blocks in the search area using Dynamic External Stop Search (DESS) algorithm. Additionally, more savings in computations are achieved in the early stage of the proposed FDS using both early search termination and adaptive pattern selection techniques. Compared to the conventional Full Search (FS) algorithm and DS algorithm, respectively, the proposed FDS algorithm achieves up to 99% and 20% more reduction in computations. The FDS algorithm guarantees high computational savings and keeping low degradation in both the Peak Signal to Noise Ratio (PSNR) and the bit-rate. Additionally, falling in a local minimum while computing the SAD is disappeared using the proposed FDS.

Keywords: DiamondSearch Algorithm, Motion Estimation

1. Introduction

ITU and ISO [1]–[4] jointly developed the H.264/AVC standard [5] for encoding the video signals in the recent few years. H.264/AVC standard uses the latest innovations in video compression techniques to provide incredible video quality and achieving up to 50% savings in the bit-rate compared to other previous standards. H.264/AVC is used over a wide range of applications such as multimedia messaging, HD-DVD, video conferencing, broadcasting, streaming, and video-on-demand. Many exhaustive tools are developed in H.264/AVC for achieving a high compression gain. Multiple reference frames, half-pel and quarter-pel accurate motion estimation, and variable block sizes are examples of such tools. Consequently, there is an increase of both the coding performance and the computational complexity of the encoding process. Block Based Motion Estimation (BB-ME) is used mainly for the motion estimation process of most existing video coding standards [1]–[4]. BB-ME is mainly used to reduce the temporal redundancy between frames. However, it can use as much as 40-80% of the total encoding time [6]. Full Search (FS) algorithm is the most well known BB-ME algorithm and it is used for most video standards [1]–[4]. Although the conventional Full Search (FS) algorithm achieves the best quality amongst various Motion Estimation (ME) algorithms and it is straightforward and has been successfully implemented on VLSI chips [7, 8], its computational complexity is very high [6]. In contrast, real time and portable multimedia devices require ultra computationally efficient video codec designs that allow for a robust and reliable video quality.

Many sub-optimal but faster ME techniques have been proposed to tackle the previous ME computational complexity problem. Some techniques are based on reducing the number of search points in the search area [9]. Although this technique reduces the computational complexity, there will be degradation in the PSNR. New Three Step Search (N3SS) [10], Four Step Search (FSS) [11], Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) [12], Hexagon Based Search (HEXBS) [13], Diamond Search (DS) [14, 15], Predictive Motion Vector Field Adaptive Search Technique (PMVFAST) [16], and Cross Diamond Search (CDS) [17] are examples for such techniques. Some other fast motion estimation techniques reduce the number of search points in the search area. Adaptive Search Window Size (ASWS) algorithm [18, 19] is a good example for such techniques. The main idea of such technique is to adaptively diminish the search window size according to the expected motion activity of the current block. If it is high, a large

[1] *Permanent address is Electronics and Communications Engineering Department – Faculty of Engineering – Mansoura University – Mansoura – Egypt.*

window size is used. Otherwise for low activity motion, a small window size will be used. The motion activity of a block is decided using offline parameters as in [18] or online parameters integrated into a model equation as in [19]. The used technique in [19] is much better than the one used in [18] and achieve better coding performance (PSNR and bit-rate) due to the high accuracy in calculating the motion activity of the current block. The main challenge in all of the previous technique is to make them suitable for VLSI implementation. Some other techniques based on simplifying or easing the matching criteria (SAD) using spatial and/or temporal Macro Blocks' (MB) features. Partial Distortion Elimination algorithm (PDE) [20]–[22] and Successive Elimination Algorithm (SEA) [23]-[29] are examples of such techniques. The advantage of such approaches is their capability to reduce the computations of the ME process with high coding efficiency.

A novel Fast Diamond Search algorithm(FDS) is implemented in this journal achieving both higher computational savings and higher coding efficiency of the transmitted video signal compared to the conventional Diamond Search algorithm. Three main steps are followed in the proposed FDS for achieving great reduction in computations compared to the state of the art fast Motion Estimation techniques. The first step is to accurately terminate the search in the early stage or adaptively selecting between either a Small Diamond Search Pattern (SDSP) or a Large Diamond Search Pattern (LDSP). This step is very effective in case of encoding video sequences with small and medium motion activities since most of the time the SDSP is selected. Consequently, high reduction in computations is achieved. The second step is to skip the unexpected candidate blocks in the search area using the DESS algorithm. The third step is to skip the redundant internal operation while calculating the SAD operation within the block using the DISS algorithm. Both the proposed DESS and the DISS use accurate adaptive (dynamic) models that adaptively changed according to some pre-estimated parameters. High reduction in computations are achieved using the proposed dynamic models with an acceptable degradation in the coding efficiency compared to the conventional Full Search (FS) and Diamond Search (DS) algorithms as well as the state of the art fast Motion Estimation techniques. It is worth mentioning that partial results of this work were implemented in [32].

The paper is organized as follows: section 2 discusses more details about the proposed FastDiamond Search (FDS) algorithm. Comparing the accuracy of the proposed FDS algorithm with the state of the art fast Motion Estimation algorithms is discussed in section 3.Conclusion is drawn in section 4.

2. FAST DIAMOND SEARCH (FDS) ALGORITHM

An important property inmost video sequences is the smoothness of the motion field for consecutive frames. As a result of such property, the Best Match Motion Vectors (BMMVs) may be allocated most of the time to be very close to or located at the center of the search area.As an illustrative experiment, the distribution of the BMMVs for six different video sequences is estimated as seen in Fig. 1. Defining the motion activity as the degree of motion of a block with respect to the center of the search area, these video sequences can be classified into different motion activity categories; i. e., low, medium, and high motion activities. As noted from Fig. 1, the BMMVs for the low motion activity video sequences (Akiyo, Coast-Guard, and News video sequences) are allocated at the center of the search area most of the time.We call this phenomenon as zero bias property. Additionally, for medium and high motion activity video sequences(Mobile, Foreman, and Football video sequences),the BMMVs are moved away from the center of the search area. We call this phenomenon as center bias property. We will benefit from both phenomena in reducing the computations as will be seen latter.

The zero bias property is used to decide if the BMMV is zero or not. It means that center of the search area is considered as the BMMV. Consequently, no need to perform Motion Estimation process and so lot of savings in computations is performed. This will be achieved by using a Dynamic Early Stop Search threshold (T_{DESS}). This threshold is dynamically adapted according to some pre-estimated parameters of the current block. The center bias property is very beneficial in reducing the computations of the DS algorithm. This is achieved by using threshold T_p to dynamically select between two patterns (either SDSP or LDSP). More reductions while performing the Motion Estimation using the previous two patterns are achieved using both DISS and DESS techniques as will be discussed soon.

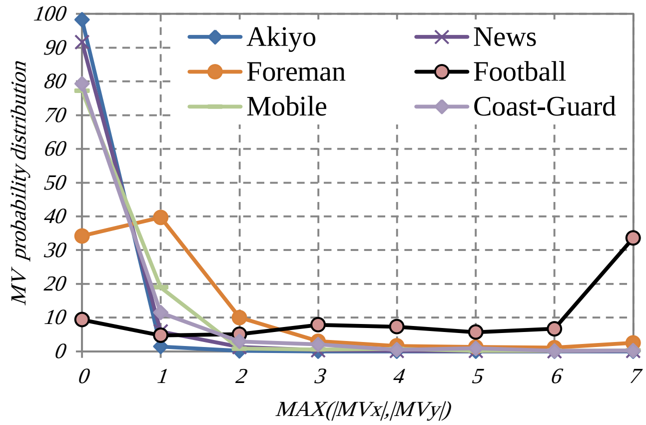

Figure 1. Distribution offor different video sequences and using FS algorithm

Fig. 2.a and Fig.2.b illustrate the two patterns used for DS algorithm. SDSP is used when the motion activity is small. The LDSP is used when the motion activity is large. Although the use of the SDSP is reducing the computations, it may degrade the coding efficiency if used at the beginning of the search process of high and medium motion activity video sequences. This is due to the possibility of falling into local minima.Using the LDSP may avoid such problem, but the computational complexity sill increase since more points will be searched. We use an adaptive threshold $T_P$to accurately decide the used pattern according to the motion activity of the current block. The motion activity of a block may be expected from the previously encoded BMMVs of surrounding blocks since there is a high correlation between such blocks.The SDSP is used if the expected motion activity is low. Otherwise, an LDSP is used. The threshold T_P is calculated as seen in Eq.1.

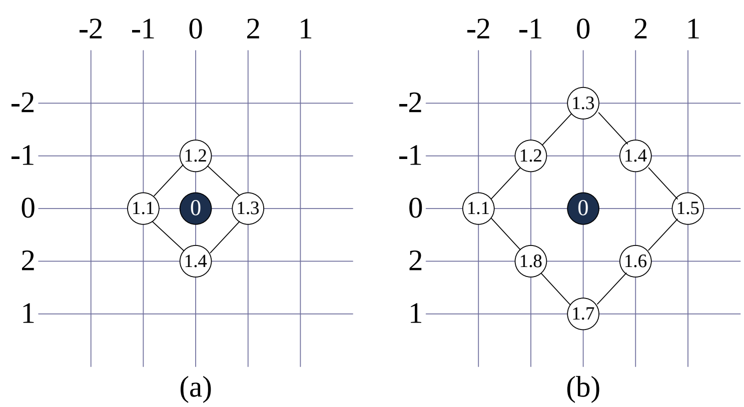

Figure 2. (a) Small Diamond Search Pattern (SDSP) (b)Large Diamond Search Pattern(LDSP)

$$T_P = \mathrm{Median}\{BMMV_{min}(m)\}, \quad m = 1,2,\ldots,K \qquad (1)$$

Where $BMMV_{min}$ is the minimum BMMVs of the neighboring blocks m. The small value of $T_P$means that the best match candidate block is located very close to the search center. A large value of T_P means that the best match candidate block is far away from the search center.

Given that the search window size is (-Δ, Δ) and the displacement with respect to the current block located at (u, v) be (x, y). The SAD between the current block in frame n and the candidate block in frame n-Ψ is:

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|I_n(u,v) - I_{n-\Psi}(u+x, v+y)| \qquad (2)$$

Where $-\Delta \leq x, y \leq \Delta$, N is the block size, and I is the pixel intensity. The proposed FDS algorithm is summarized in the following sub-sections.

A. Initial Step (Dynamic Early Stop Search)

For low motion activity video sequences, the probability of having the best mach candidate block at the center of the search area is very high. Consequently, the $SAD_{curr}(i)$ between the current block and the candidate block ilocated at the center of the search areais estimated. The thresholds T_{DESS} (its derivation given in section 3.B) and T_P(Eq.1) are calculated. If $SAD_{curr}(i) \leq T_{DESS}$, then the search will stop immediately and outputs the candidate block i located at the center of the search area as the best match candidate block as seen in Fig.3.a. Otherwise, go to step B.

B. The Decision of Using Small Diamond Search Pattern (SDSP) or Large Diamond Search Pattern (LDSP)

If the center of the search area is not selected as BBMV, further calculations should be calculated. The threshold $T_P$is used to select the initial diamond pattern (either SDSP or LDSP) using the following condition. If $T_P \leq 1$, go to step C. Otherwise, go to step D.

C. Small Diamond Search Pattern (SDSP)

The SDSP in Fig.3.b is used for calculating the best match candidate block. All the points in the SDSP will be checked one by one against the minimum SAD so far. Both the DISS and the DESStechniques will be used to speed up the ME process by eliminating unnecessary computations as will be discussed in section *3-A* and section *3-B*. If the best match candidate block is not obtained after searching all the points in the SDSP, then check the search point with the minimum SAD. If it is located at the center of the SDSP, then safely stop the search. Otherwise, use this point as an ISC and repeat step *C* (see Fig.3.c).

D. Large Diamond Search Pattern (LDSP)

The LDSP consists of eight points as shown in Fig.2.b. These points are checked one by one to calculate the optimum SAD point and using of both the DISS and DESS techniques. If the best match candidate block is not reached after checking all the point of the LDSP, the search point with the minimum SAD is checked. If this point is not located at the ISC, then start step D over again and consider the point with minimum SAD as the new search center for the new LDSP. If the point with the minimum SAD is located at the center of the LDSP, an SDSP is used as a final stage. If the best match is not caught by the procedure, then the point with the minimum SAD will be the best match candidate block as seen in Fig.3.d. Due to the high accuracy of the proposed thresholds, the optimum candidate best match can not fall into local minima as may happen in case of using the conventional DS algorithm.

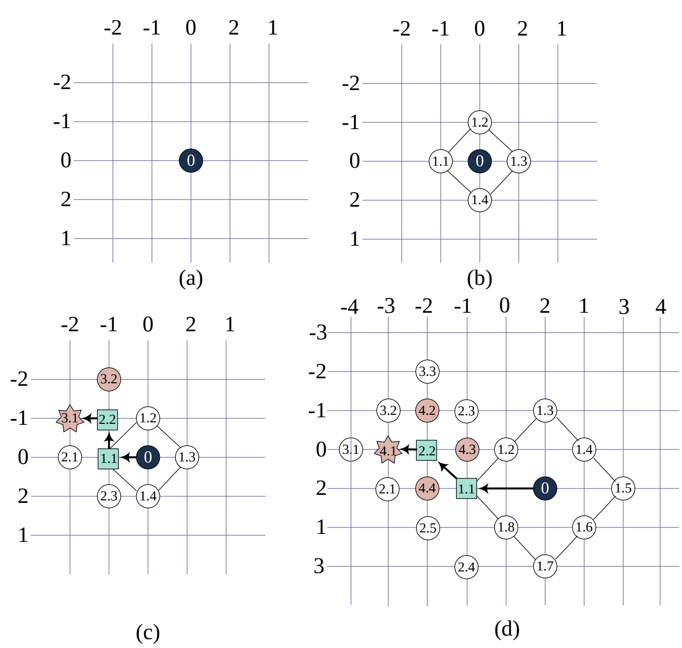

Figure 3. Fast Diamond Search (FDS) Algorithm

3. The Proposed Early stop search termination Algorithms

The computational complexity *CC*of the block based Motion Estimation technique is approximated as follows:

$$CC = [l_1 \times l_2 \,.\, (Sub + Abs + Add)].\,S \qquad (3)$$

Where l_1 and l_2 are the rows and columns of both the current and the candidates blocks, respectively.*S* is the number of search points in the search area.*Sub*, *Abs*, and *Add* are the number of subtraction, absolute value, and addition operations required for calculating the Sum of Absolute Difference (SAD) in Motion Estimation process. In this work, the computations of Eq.3 are reduced by reducing the *Sub*, *Abs*, and *Add* operations per candidate block using the proposed*DISS*technique. Additionally, more reduction in the number of the search point *S*is achieved using the proposed*DESS*. The Early Search Termination algorithm is summarized in the following sub-sections.

A. Dynamic Internal Stop Search Algorithm (DISS)

The proposed *DISS*algorithm reduces the computations of the *ME* process by reducing the SAD operations (i. e., reducing *Sub*, *Abs*, and *Add* operations) that are required for getting the best match candidate block. Consequently, the term $[l_1 \times l_2 . (Sub + Abs + Add)]$ in Eq.3 is reduced. The following four steps summarize the proposed *DISS* algorithm.

- ***Step (i)*:** Divide both the current and the candidate blocks into groups of pixels. Each group contains a number of rows (2^l), where$l = 0,1,2, \ldots , l_1/4$
- ***Step (ii)*:** The *SAD* value will be partially accumulated starting with the first group of both the current and the candidate blocks.
- ***Step (iii)*:** The partially accumulated *SAD* value (*PSAD*) is compared to a pre-determined dynamic threshold T_{DISS}.
- ***Step (iv)*:** If the partially estimated value of$PSAD > T_{DISS}$, then stop accumulating *PSAD* for further groups and proceed to the next candidate block in the search area. Otherwise, continue accumulating *PSAD* by adding the next group of pixels to the previous *PSAD* and update the threshold accordingly.

The threshold T_{DISS} depends on the normalized minimum *SAD* of the scanned blocks so far in the search area (*SADmin-curr*/$[l_1 \times l_2]$). The threshold T_{DISS}can be estimated as follows:

$$T_{DISS}(j) = j \times P \times \frac{SAD_{min-curr}}{l_1 \times l_2} \qquad (4)$$

Where j is the group index and P is the number of pixels per group. $SAD_{min-curr}$ is the minimum SAD of the scanned blocks so far in the search area.

Local minima may result from falsely skipping some scanned blocks of the search area in the early stages of the algorithm. It was noticed experimentally that if the PSAD of the first group(s) for a candidate block is greater than the calculated threshold $T_{DISS}(j)$, the candidate block is skipped. Although, if we further continue accumulating the PSAD for the next group(s) in that block, the final accumulated PSAD might not exceed the threshold $T_{DISS}(j)$. This may result in falling in a local minima problem. To avoid this problem, an accelerator parameter **Ω** is added to the threshold in Eq.4 to control the rate of the PSAD skipping operation in the candidate block. The value of **Ω** parameter is illustrated in Eq.5:

$$\Omega = \varepsilon \times \frac{SAD_{min-curr}}{l_1 \times l_2}, \quad \varepsilon = 1, 2, \ldots, P/2 \qquad (5)$$

Increasing the value of Ω will make it harder to skip the PSAD calculations for a candidate block in the early stage of the PSAD calculations. Consequently, decreasing the possibility of falling intothe local minima problem.However, this is against our target here as it might increase the non-skipped blocks and hence increasing the computational complexity of the Motion Estimation process.

Exhaustive experiments reveal that the false skipping of PSAD operations (Local minima problem) is not a dominant case for all the candidate blocks in the search area. Additionally, these false skipping operations depend mainly on the choice of the initial group to be partially accumulated for calculating PSAD. As a result of previous discussions, the effect of the acceleration parameter Ω should be relaxed with the further accumulating of PSAD from one group to the next one. This is achieved by subtracting a relaxation parameter θ from the total threshold T_{DISS} in order to relax the effect of adding the accelerator parameter Ω. The value of θ is illustrated in Eq.6 where N is the total number of the groups in a block. At the last group of a block, Ω will be completely eliminated by θ and T_{DISS} will settle to the value of $SAD_{min-curr}$. The final form of the proposed DISS threshold is given in Eq.7.

$$\theta = \frac{(j-1) \times \Omega}{(N-1)} \qquad (6)$$

$$T_{DISS}(j) = j \times P \times \frac{SAD_{min-curr}}{l_1 \times l_2} + \Omega - \frac{(j-1) \times \Omega}{(N-1)} \qquad (7)$$

B. Dynamic External Stop Search Algorithm (DESS)

The DESS algorithm is trying to reduce the computations of the Motion Estimation algorithm given in Eq.3. This is achieved by reducing the candidate blocks in the search areaif they cannot be considered as a best match candidate block with a minimum SAD so far. The DESS algorithm is considered as an additional tool for reducing the computations is the DISS algorithm fails to skip internal operations of PSAD. If the PSAD calculations of a candidate block are not skipped, then the value of the$SAD_{min-curr}$for the current block will be updated according to the value of the obtained final PSAD for that candidate block. If the final PSAD is less than the $SAD_{min-curr}$, then the value of the $SAD_{min-curr}$ is replaced by the current value of the final PSAD. Thereafter, we check this updated $SAD_{min-curr}$ against a pre-determined Dynamic External Stop Search threshold $T_{DESS}(i)$; where i is the index of the candidate block in the search window. If the updated $SAD_{min-curr} \leq T_{DESS}(i)$, then skip all the remaining candidates in the search window and select the i block as our best candidate block.

Exhaustive experiments reveal that the best match block in the search window has a minimum SAD that is highly correlated with the minimum SADs of the neighbors of the current block, i.e., blocks 1, 2, 3, and 4 in Fig.4. This is due to the high correlation property of BMMVs of neighboring blocks in one frame [19]. Therefore, the minimum SADs of the surrounding blocks to the current block can be used to calculate the function £ of Eq.8 to form the threshold $T_{DESS}(i)$. A small register bank is required to store the minimum SADs of the surrounding blocks so far. The function £ can be simply set to the average of the minimum SADs of the surrounding blocks to the current block (i.e., blocks 1, 2, 3, and 4 in Fig.4). An improvement in the accuracy of the threshold $T_{DESS}(i)$ can be achieved by using the median function instead of using the average value. Using the median function is considered as the average of the best two neighboring blocks since it will exclude the irregular minimum SAD values of the neighbors to the current block from the calculations.

$$T_{DESS}(i) = \mu_1 \times £ + \mu_2 \qquad (8)$$

Where £ is defined as:

$$£ = \begin{cases} \frac{1}{N}\sum_{m=1}^{N} SAD_{min}(m) \quad \text{or,} \\ \text{Median}\{SAD_{min}(m)\}, \quad m = 1, 2, \ldots, K \end{cases} \qquad (9)$$

Where K is the number of the surrounding coded blocks so far, μ_1 is an accelerator parameter, and μ_2 is a constant factor set experimentally to zero. It is worth mentioning that the threshold in Eq.8 is computed only once per a current block. The accelerator parameter μ_1 is set experimentally to 0.75. Fig.5 illustrates the whole flow diagram of the proposed DISS and DESS algorithms combined together.

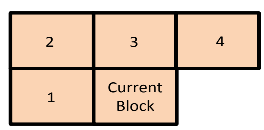

Figure 4. The used four surrounding blocks in calculating $T_{DESS}(i)$

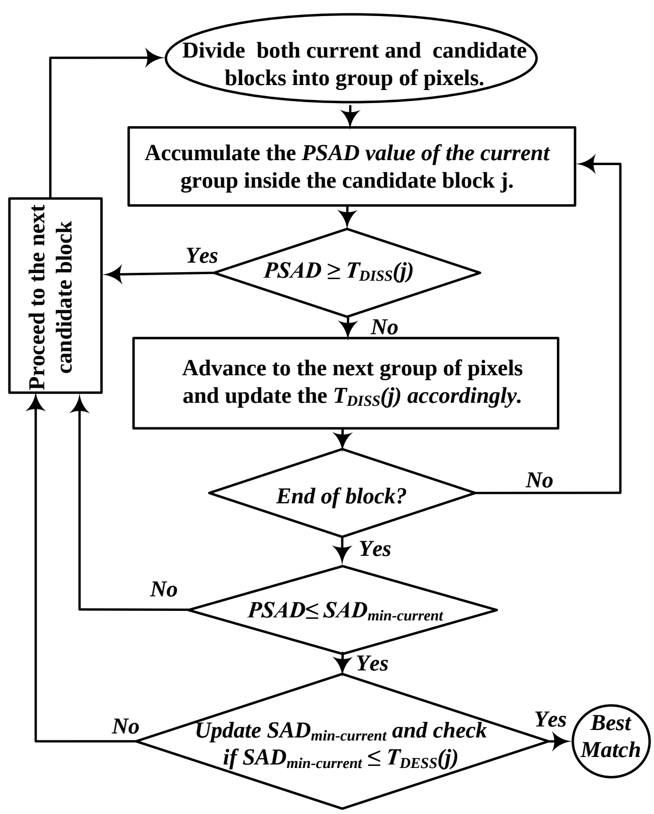

Figure 5. Complete discription of DISS and DESStechniques

4. EXPERIMENTAL RESULTS AND DISCUSSIONS

The effectiveness of the proposed FDS algorithm is evaluated using the JM12.4 reference software [16].Performance measurement is based on three main parameters:

1. The PSNR difference (ΔPSNR) of the reconstructed video sequence.
2. The increase in the bit-rate percentage (ΔBR%).
3. The ME time saving percentage (METS%).

These three parameters are measured with respect to the conventional FS algorithm. Four different video sequences with different motion activities are tested. Theselection of the BMMV is based on the luminance component of a block of size 16×16. The search area is of size 32×32 pixels. The proposed FDS algorithm is compared to FS, N3SS, 4SS, PMVFAST, HEXBS, DS, and CDS algorithms. From table 1, it is notice that the Motion Estimation Time Saving (METS) of the proposed FDS is significant compared to other state of the art fast ME techniques. Basically, it achieves approximately 99% and 20% saving in ME time, for high motion activity video sequence (e. g., Football video sequence), and compared to the conventional FS and DS, respectively. This high saving in computations combined with small degradation in both the PSNR and bit-rate of the transmitted video sequence.

The average number of Absolute Difference (AB) operations per MV for Foreman and Akiyo video sequences is presented in Fig.6 and Fig. 7, respectively. Compared with the other techniques, the proposed FDS algorithm has the lowest AB operations. This reflects the superior performance of the proposed FDS algorithm that will lead to speed up the process of ME. In addition, the performance of the proposed algorithm is measured using the Rate Distortion (RD) curves [31] (calculated at 30 frames per sec) as shown in Fig.8 and Fig. 9. It is clear from both Figures that the proposed FDS performs better than CDS algorithm and very close to the DS algorithm.

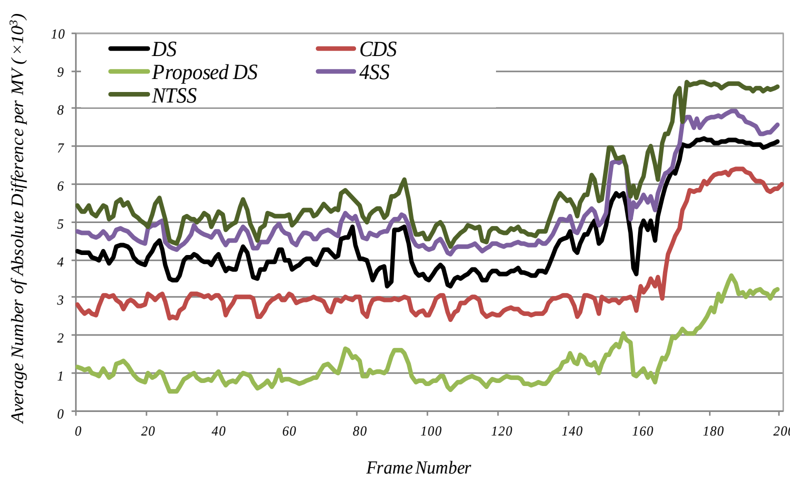

Figure 6. Average number of absolute difference per MV using Foreman video sequence

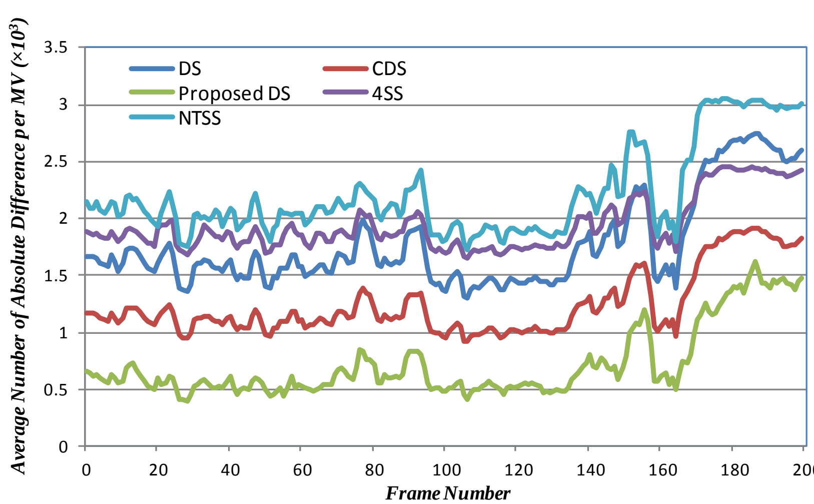

Figure 7. Average number of absolute difference per MV using Akiyo video sequence

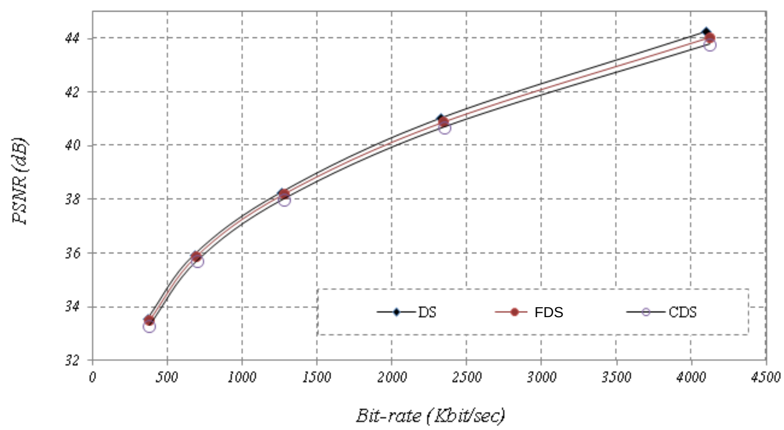

Figure 8. Bit-rate Distortion Curves comparison using Foreman video sequence

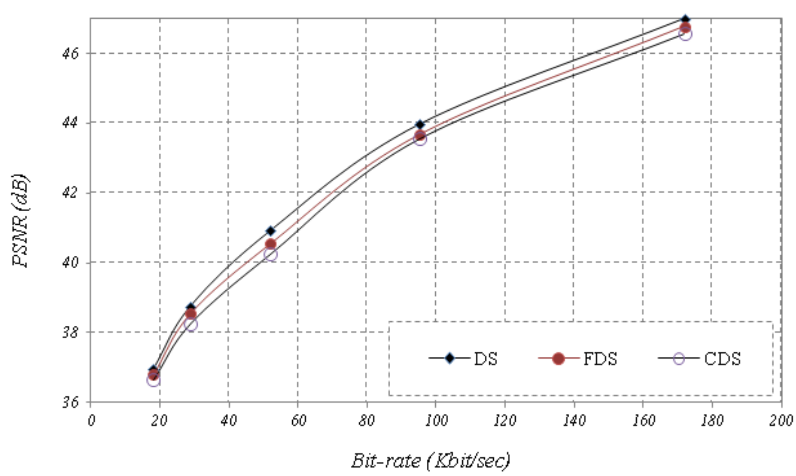

Figure 9. Bit-rate Distortion Curves comparison usingAkiyo video sequence

TABLE I. SYMULATION AND COMPARISON TABLE

ME technique	Akiyo (QCIF)			Forman (CIF)			Coast-Guard (QCIF)			Football (CIF)		
	PSNR(dB)	BR	METS %	PSNR(dB)	BR%	METS	PSNR(dB)	BR	METS	PSNR(dB)	BR	METS
FS	38.55	28650	194.796	37.71	474840	1143.653	34.79	243900	671.30	37.10	1036980	1021.45
	ΔPSNR(dB)	BR %	METS %	ΔPSNR(dB)	BR%	METS %	ΔPSNR(dB)	BR%	METS %	ΔPSNR(dB)	BR%	METS %
N3SS [10]	-0.01	0.09	74.62	-0.4	0.81	70.04	-0.04	0.08	72.16	-0.62	0.33	65.08
4SS [11]	0.00	0.05	79.93	-0.2	0.43	74.64	-0.01	0.06	77.23	-0.51	0.29	73.95
PMVFAST[12]	-0.06	-0.21	85.71	-0.05	-2.06	95.13	-0.01	0.05	93.83	-0.01	0.62	94.38
HEXBS [13]	0.00	0.03	95.01	-0.02	0.98	94.56	0.00	0.04	95.18	-0.03	1.44	94.62
DS [14]	-0.06	-0.31	85.73	-0.03	-1.88	82.12	0.00	0.19	87.78	-0.02	0.67	80.42
CDS [15]	-0.05	0.21	89.11	-0.05	1.98	88.63	-0.07	2.64	92.13	-0.04	2.55	84.52
FDS	-0.01	0.17	99.37	-0.03	2.32	99.15	0.00	2.29	99.43	0.00	2.02	99.34

5. CONCLUSION

A novelFast Diamond Search (FDS) algorithm that accurately reduces computations is proposed. Efficient Early stop search termination algorithms are inherently implemented with the FDS algorithm for a superior reduction in computations.Complexity reduction is achieved using two main smart models to skip the unnecessary redundant internal SAD operations and also to skip the irrelevant blocks' operations in the search area. Additionally, more savings in computations are achieved in the early stage of the proposed FDS using both early search termination and adaptive pattern selection techniques. There is approximately 99% and 20% Motion Estimation Time Savings (MEST) compared to both the conventional FS and the conventional DS, respectively. These savings combined with an acceptable degradation in the transmitted video coding efficiency.Consequently, the proposed FDS algorithm is appropriate for real time video applications such as video HD video podcasting and video conferencing over wireless networks.

ACKNOWLEDGMENT

The authors acknowledge the support of the computer Engineering Department - University of Bahrain – Bahrain for their support to finalize this work.

REFERENCES

[1] ITU-T Rec. H.263, "Video coding for low bit rate communication," 1998.

[2] ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC, " Advanced video coding for generic audiovisual services," 2003.

[3] ISO/IEC CD 13818-2 – ITU-T H.262 (MPEG-2 Video), " Information technology – Generic coding of moving pictures and associated audio information: Video, " 1995.

[4] ISO/IEC 14496-2 (MPEG-4 Video), " Information technology – Coding of audio visual objects," 1999.

[5] T. Wiegand, G. J. Sullivan, G. Bjntegaard, A. Luthra, "Overview of the H.264/AVC video coding standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp.560-576, July 2003.

[6] W. I. Chong, B. Jeon, and J. Jeong, "Fast motion estimation with modified diamond search for variable motion block sizes," in Proceedings of the International Conference on Image Processing, 2003, Vol.3, pp. II- 371-4 14-17, 14-17 Sept. 2003.

[7] S. Goel, Y. Ismail, P. Devulapalli, J. McNeely, and M. Bayoumi, "An Efficient Data Reuse Motion Estimation Engine," IEEE Workshop on Signal Processing Systems (SIPS 06), Banff, Canada, Sept. 2006.

[8] J. -C. Tuan, T. -S. Chang, and C. -W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," IEEE Trans. Circuits Syst. Video Technol., Vol. 12, pp. 61-72, Jan. 2002.

[9] G.-L. Li and M.-J. Chen, "Fast Motion Estimation Algorithm by Finite-State Side Match for H.264 Video Coding Standard," in IEEE Asia Pacific Conference on Circuits and Systems, APCCAS'06. , pp. 414-417, 2006.

[10] R. Li, B. Zeng, and M. L. Liou, "A New Three-Step Search Algorithm For Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438-442, Aug. 1994.

[11] L. M. Po and W. C. Ma, "A Novel Four-Step Search Algorithm For Fast Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313-317, June 1996.

[12] Alexis M. Tourapis, Oscar C. Au, and Ming L. Liou, "Highly Efficient Predictive Zonal Algorithms for Fast Block-Matching Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 12, NO. 10, October 2002

[13] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-Based Search Pattern For Fast Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol.12, pp. 349-355, May 2002.

[14] S. Zhu and K.-K. Ma, "A New Diamond Search Algorithm For Fast Block Matching Motion Estimation," IEEE Trans. Image Processing, vol. 9, pp. 287-290, Feb. 2000.

[15] Luheng Jia; Au, Oscar C.; Chi-ying Tsu; Yongfang Shi; Rui Ma; Hong Zhang, "A diamond search window based adaptive search range algorithm," Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on , vol., no., pp.1,4, 15-19 July 2013.

[16] Joint Video Team, "Reference Software JM12.4," http://iphome.hhi.de/suehring/tml/download/".

[17] C. H. Cheung and L. M. Po, "A Novel Cross-Diamond Search Algorithm For Fast Block Motion Estimation," IEEE Trans. Circuits Syst. Video Technol., vol. 12, pp. 1168-1177, Dec. 2002.

[18] Sumeer Goel, Yasser Ismail, and Magdy A. Bayoumi, " A Fast Block-Matching Algorithm and Its VLSI Architecture for Real-Time Video Coding," Oxford Journals - The Computer Journal (2012) 55(1): 35-46 first published online April 29, 2011.

[19] Yasser Ismail, Jason McNeely, Mohsen Shaaban, and Magdy A. Bayoumi, "Fast Motion Estimation Algorithm Using Dynamic Models for H.264 Video Coding," IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), Volume 22, Issue 1, pp. 28 – 42, January 2012.

[20] Digital Video Coding Group, ITU-T Recommendation H.264 Software Implementation, Telenor R&D, 1995.

[21] S. Eckart and C. Fogg, "ISO/IEC MPEG-2 Software Video Codec," Proc. SPIE, vol. 2419, pp. 100-118, 1995.

[22] Xuan Jing, Lap-Pui Chau, "Partial Distortion Search Algorithm Using Predictive Search Area for Fast Full-Search Motion Estimation," IEEE Signal Processing Letters, vol. 14, No. 11, pp. 840 – 843, Nov. 2007.

[23] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," IEEE Trans. Image Processing, vol. 4, pp. 105–107, Jan. 1995.

[24] X. Q. Gao, C. J. Duanmu, and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," IEEE Trans.Image Processing, vol. 9, pp. 501–504, Mar. 2000.

[25] M. Brünig and W. Niehsen, "Fast full-search block matching," IEEE Trans. Circuits Syst. Video Technol., vol. 11, pp. 241–247, Feb. 2001.

[26] Tae Gyoung Ahn, Yong Ho Moon, and Jae Ho Kim, "Fast Full-Search Motion Estimation Based On Multilevel Successive Elimination Algorithm" IEEE Trans. on Circuits and Systems for Video Technology, vol. 14, No. 11, pp. 1265 – 1269, Nov. 2004.

[27] Yu-Wen Huang, Shao-Yi Chien, Bing-Yu Hsieh, and Liang-Gee Chen, " Global Elimination Algorithm and Architecture Design for Fast Block Matching Motion Estimation," IEEE Trans. on Circuits and Systems for Video Technology, vol. 14, No. 6, June 2004.

[28] Chen, W.-G., Ling, Y., "Noise Variance Adaptive Successive Elimination Algorithm For Block Motion Estimation: Application For Video Surveillance," Signal Processing, IET, vol. 1, No. 3, pp. 150 – 155, Nov. 2004.

[29] Ce Zhu, Wei-Song Qi, Ser, W., "Predictive Fine Granularity Successive Elimination For Fast Optimal Block-Matching Motion Estimation," IEEE Trans. on Image Processing, vol. 14, No. 2, Feb. 2005.

[30] S. Yang, L. Zhenyu, T. Ikenaga, and S. Goto, "Enhanced Strict Multilevel Successive Elimination Algorithm for Fast Motion Estimation," in IEEE International Symposium on Circuits and Systems, ISCAS'07., pp. 3659-3662, 2007.

[31] G. Bjotegaard, “Calculation Of Average PSNR Differences Between RD-Curves,” Video Coding Experts Group (VCEG), VCEG-M33, Thirteenth Meeting: Austin, Texax, USA, 2-4 April, 2001.

[32] Ismail, Y.; McNeelly, J.; Shaaban, M.; Bayoumi, M.A., "Enhanced efficient Diamond Search algorithm for fast block motion estimation," ISCAS 2009. IEEE International Symposium on Circuits and Systems, 2009., pp.3198,3201, 24-27 May 2009.

Dr. Yasser Ismail received the B.Sc.degree in Electronics & Communications Engineering from Mansoura University, Mansoura, Egypt, in 1999, the M.Sc. degree in Electrical Communications from Mansoura University, Mansoura, Egypt, in 2002, the M.Sc. degree in Computer Engineering from University of Louisiana at Lafayette, Louisiana, USA, in 2007. Dr. Yasser Ismail got his Ph.D. from the University of Louisiana at Lafayette in May 2010. Dr. Yasser Ismail worked as an assistant professor in Umm Alqura University – KSA from 2010 to 2012. He is currently working as an assistant professor in University Of Bahrain (UOB) - Bahrain. Dr. Yasser permanently working at the Electronics and Communications Engineering Department – Faculty of Engineering – Mansoura University – Mansoura – Egypt. Dr. Yasser is served as a reviewer for several conferences and journals, including ISCAS 2010, ICIP 2010, ICIP 2011, ICECS2013, Transaction on Circuit and System for Video Technology (TCSVT), and IEEE Transactions on Image Processing, and Signal Processing. He has also gained many valuable projects from KSA, NSF, and Bahrain. Dr. Yasser served in the organizing committee of 2013 IEEE International Conference on Electronics, Circuits, and Systems (ICECS2013). His research of interest includes video processing, digital signal processing, Robotics, RFID, Localization, VLSI, FPGA, wireless communication systems, and low power embedded systems.