Journal of Engineering Technology Vol. 3(1): 18-22, 2013
ISSN 2231-8798

A Novel Search Pattern for Motion Estimation of Video Encoder

Noor Hidayah M.Yunus[1], L.A.Tiron[2], N.Rodziah A.Ghafar[3], P.D.Abd.Aziz[4] & F.Z.Hamidon[5]
[1,2,3] Section of Communication Technology
[4,5]Section of Electrical Technology
Universiti Kuala Lumpur British Malaysian Institute

Corresponding email: noorhidayahm@bmi.unikl.edu.my

Abstract: With the improvement in image sequence compression or video coding is a normal extension of the research in image compression or coding lively over several decades. One of the exciting prospects of the improvement in video compression is that multimedia information comprising image, video, and audio has the potential to become just a different data type. For such data to be persistent, it is essential that the data encoding be standard across different platforms and applications. Beyond that, Motion Estimation (ME) technique is a very important function in video compression standard and the ME process introduces very high computational obligation for the video encoder if the usual Full-Search (FS) method is used. Numerous fast ME algorithms have been proposed to deal with this problem. This project proposes the improvement of ADLISP to reduce computational complexity and to obtain significantly better PSNR performance. Experimental results exhibit good PSNR performance and the average number of search points of the proposed algorithm compared with several other popular search algorithms.

Keywords: Block matching, complexity, full-search, macroblock, motion vector, video compression.

1.0 INTRODUCTION

With the advancement in video compression technology, ISO/IEC MPEG and ITU-T VCEG have jointly developed a new video coding standard, known as H.264/AVC. H.264/AVC can achieve about twice coding efficiency higher than MPEG-4. Motion Estimation (ME) is an important part of H.264/AVC encoding progress with high computational complexity [1]. H.264/AVC implements the block based ME in which involves fast search algorithms and block matching methods. Its fundamental idea is to segment the current frame and the reference frame into sub-blocks.

The MVs for all the pixels in one block are treated as one single vector to be estimated. This kind of block-based motion compensation is widely adopted by video coding standards [2-4] due to its simplicity. The block-based MVs can be estimated by using block matching, in which minimizes a measure of matching error. The matching error between the block at position (x, y) in the current image, I_t, and the candidate block at position $(x+u, y+v)$ in the reference image, I_{t-1}, mostly the block-based motion estimation algorithms are based on computing the Sum of Absolute Differences (SAD) between corresponding elements in the candidate and reference blocks.

$$SAD_{(x,y)}(u,v) \equiv \sum_{j=1}^{N}\sum_{i=1}^{N} \left| I_t(x+i, y+j) - I_{t-1}(x+u+i, y+v+j) \right|, \; [5] \qquad (1)$$

Where the number of pixels per block is N. The absolute difference of all the pixels corresponding to the current and reference blocks are computed in parallel. In this (1), the best estimate of the MV, $(\acute{u}, \acute{v})$ is defined to be the (u, v) which minimizes $SAD_{(x,y)}(u, v)$. This estimate, $(\acute{u}, \acute{v})$, can be obtained by using the Full-Search (FS) algorithm which calculates and compares the SADs for all the search positions, $\{(x + u, y + v)\}$ in the reference image, I_{t-1} [5]. FS estimation is guaranteed to find the minimum SAD in the search window but it is computationally extensive. Hence, FS algorithm has high precision, but it may induce high computational complexity. Now many FS algorithms have been proposed [9-14] in which has low computational complexity but closely approaches the performance of FS algorithm.

Journal of Engineering Technology Vol. 3(1): 18-22, 2013
ISSN 2231-8798

The rest of the paper is organized as follows; section 2.0 describes the various fast ME algorithms strategy and the proposed ADLISP-A algorithm besides determination concept of the PMV. Computer simulations are used to evaluate the performance of the proposed algorithm and the results are presented in section 3.0. Finally, some concluding remarks are given in section 4.0.

2.0 MOTION ESTIMATION ALGORITHMS

The commonly used ME technique in all the standard video encoder-decoder is the Block Matching Algorithm (BMA). BMA was been widely accepted by many video coding standard in such way that to reduce the temporal redundancy between different frames and also the simplest technique to follow. FS [6] is the most computationally expensive BMA among all algorithms whilst gave the highest PSNR performance. Many fast search algorithms have been proposed to reduce the number of block matching computations and gave significantly better PSNR in which includes the well-known Three Step Search (TSS) [7], the Diamond Search (DS) [8], the Cross Diamond Hexagonal Search (CDHS) [9], the Hexagon Based Search (HEXBS) [10], the Adaptive Root Pattern Search (ARPS) [11], and the Adaptive Double Layered Initial Search Pattern (ADLISP) [12]. This algorithm is described in the following section. These algorithms also serve as the benchmark for the development of the new fast search algorithms.

A. Fast Search Algorithms

A.1. Three Step Search (TSS)

TSS [7] is one of the earliest attempts on fast BMA and the first non-full search algorithm. TSS is the method to compute motion displacements up to 6pixels/frame in which if the object is assumed to move over a 6 pixel range, it necessary to test 6 different horizontal displacements in each of 6 vertical positions. This search starts with a step size equal to or slightly larger than half of the maximum search range. In each step, nine search points which is consists of a central point of the square search and other eight search points located on the search area boundaries are considered. This proceeds by moving the search area center to locate the best matching point and the macro-block at that location is the best match as shown in Fig.1.

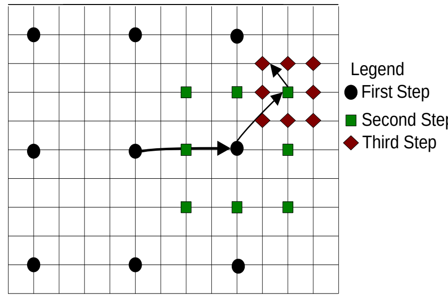
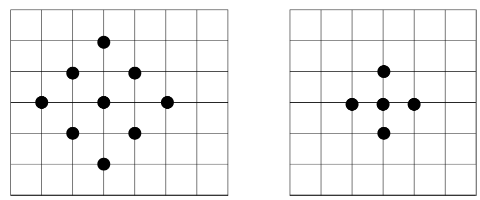


Fig.1 TSS Strategy [7]

A.2. Diamond Search (DS)

The DS implements two different types diamond-shaped search patterns [8] which is Large Diamond Search Pattern (LDSP) with nine search points and Small Diamond Search Pattern (SDSP) with five search points. These two patterns and the DS Strategy are illustrated in Fig.2 and Fig.3:



(a) LDSP (b) SDSP

Fig.2 DS patterns

The first step uses LDSP until that it reaches the edge of the search window, or a new minimum matching distortion point occurs at the center of LDSP. The consequent steps are also similar with previous LDSP, but the number of search points is checked either 3 or 5 and are illustrated in LDSP 2, LDSP 3 and LDSP 4. The last step, the search pattern is then switched to SDSP, which is used to refine the search algorithm and the location with the least weight is the best match.

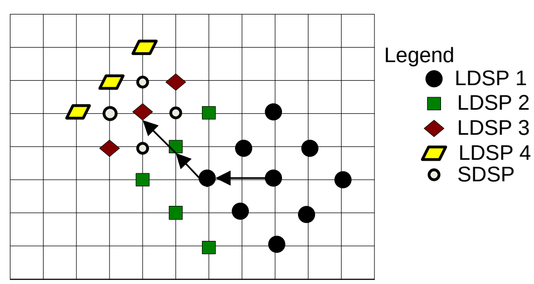


Fig.3 DS Strategy [8]

Journal of Engineering Technology Vol. 3(1): 18-22, 2013
ISSN 2231-8798

A.3. Cross Diamond Hexagonal Search (CDHS)

In the order of the basis of DS [8], CDHS [9] uses hierarchy pattern structure to improve the search efficiency and fundamentally employ two cross-shaped search patterns consecutively in the very beginning steps and switch with diamond shaped patterns. To further reduce the checking points, two pairs which are consists of orientated vertical and horizontal of hexagonal search patterns are proposed in conjunction with candidates found located at diamond corners as shown in Fig.4. Characteristically, CDHS is fast and accuracy in most cases and has additional advantage on large motion video sequences.

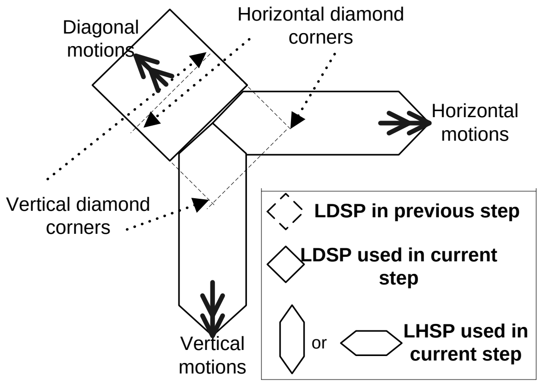


Fig.4 Search Patterns switched for different directions [9]

A.4. Hexagonal Based Search (HEXBS)

HEXBS [10] search pattern is depicted in Fig.5 and the inner search for the algorithm is carried out using the Large Hexagonal Search Pattern (LHSP) covering the point as shown by Small Hexagonal Search Pattern (SHSP) pattern.

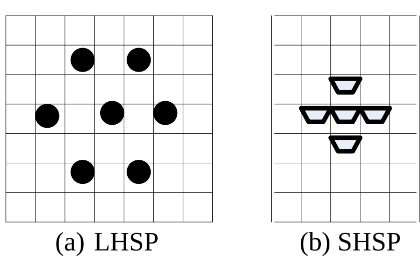


Fig.5 HEXBS search patterns

Fig.6 illustrates an example of HEXBS algorithm strategy which has shown similar step as DS, but in contrast with the DS that uses nine search points, the HEXBS adopts seven search points to achieve faster processing due to fewer search points being evaluated and works well than DS for videos with more oblique motion. HEXBS has shown significant improvement over DS algorithm.

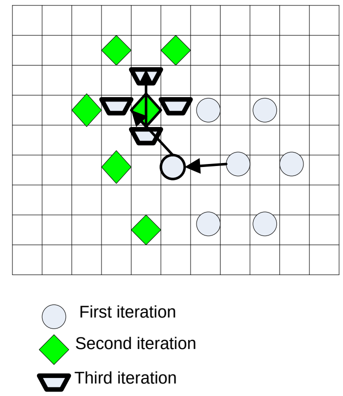


Fig.6 HEXBS Strategy [10]

A.5. Adaptive Root Pattern Search (ARPS)

Adaptive search pattern is based on inter block correlation and ARPS [11] algorithm makes use of that fact. Indeed ARPS uses the correlation between a current block and neighbor blocks to determine PMVs. When the Macro-Blocks (MBs) around the current block moved in a particular direction then there is a high probability that the current block will also have a similar MV. ARPS uses the MV of the MB to its immediate left to predict its own MV. In addition to checking the location pointed by the PMV, it also checks at a rood pattern distributed points, as shown in Fig.7, where they are at a pattern size of P = Max (|X|, |Y|). X and Y are the x-coordinate and y-coordinate of the PMV.

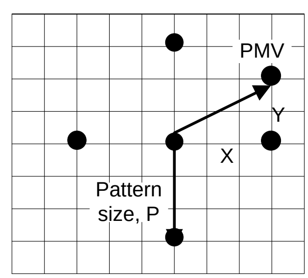


Fig.7 ARPS Strategy [11]

A.6. Adaptive Double-Layered Initial Search Pattern (ADLISP)

ADLISP [12] is a pair of complementary double-layered (inner layer and outer layer) initial search patterns. The ADLISP strategy is shown in Fig.8.

The initial search step of this algorithm consists of a pattern with eight search points which is four in both inner

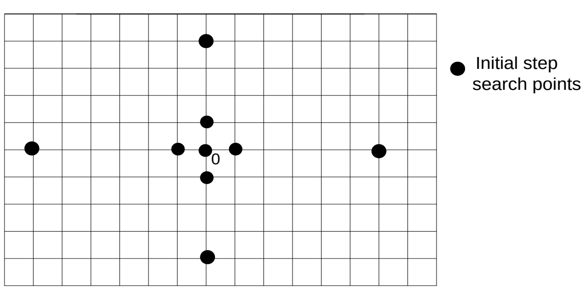
Journal of Engineering Technology Vol. 3(1): 18-22, 2013
ISSN 2231-8798

and outer. The inner layer is a SDSP search points and the outer layer is an asymmetric LCSP. Even as in Fig.8(a) gives an example that the estimated MV locates at the center of search range. After the (0, 0), (±1, 0) and (0, ±1) point is checked, by condition the best-match vector locates in the center then the outer layer search points are skipped and the search ends with five search points of inner layer.

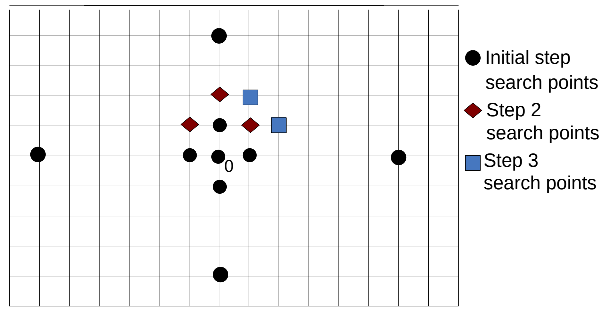
As shown in Fig.8(b), iterative SDSP is applied when the best vector of initial search falls on the inner layer of ADLISP after checking all nine initial search points. After first SDSP search, the best-matched vector moves from (0, 1) to (1, 1) which is selected after initial search step. Second SDSP search from (1, 1) does not find any best vector and consequently the search is completed with the output MV (1, 1) in the example.

In Fig.8(c), the best vector of initial search falls on (0, 4) which is one of the outer layer search point. E-HEXBS pattern search is applied from (0, 4). The best-matched vector moves from (0, 4) to (-2, 4) after first hexagon pattern search. The second hexagon pattern search from (-2, 4) does not find any best vector on the six check points, hence the search continues with 6-sidebased fast inner search. In the example, (-3, 6) and (-4, 4) have minimum distortion among the six search points of second hexagon pattern.

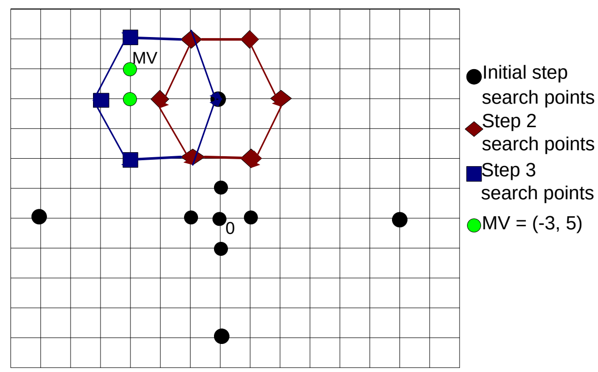
According to the hexagon with six side-based fast inner search, (-3, 4), and (-3, 5) need to be further checked [20]. The final result is the MV (-3, 5) that has lowest distortion of all searched vectors in the example. The final refinement of six-side-based fast inner search illustrated in Fig.8(d). The flow system diagram of ADLISP-A is shown in Fig.9.



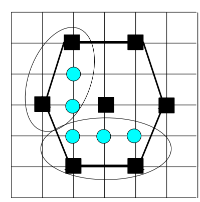
(a)



(b)



(c)



(d)

Fig.8 ADLISP Strategy [12]:

(a) First step stop with (0, 0) MV estimated $r_x = 6$ and $r_y = 4$.

(b) SDSP process when initial search result falls on inner layer

(c) 6-side fast inner search process when initial search result falls on outer layer

(d) The MV final result: check 2 or 3 points of the final refinement of E-HEXBS pattern.

A.7. The Proposed ADLISP-A Algorithm

On the beginning of the algorithm method was ME phase of each MB in which checked the Predicted MV and the nonmoving (0, 0) vector. The MVs are found by minimizing the SAD. The minimum MV is used as the center of the search range. The initial search pattern is will be applied after the search center is set.

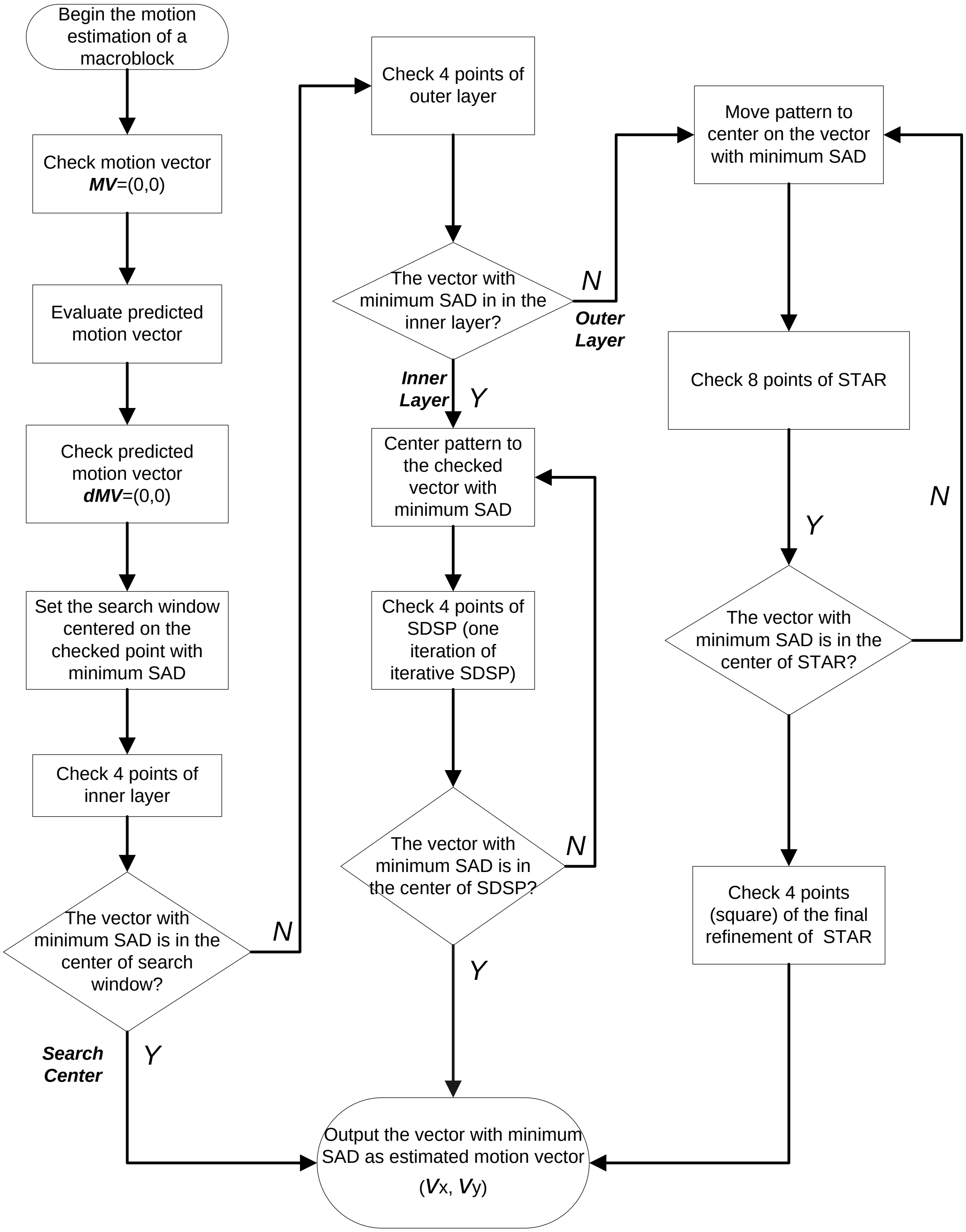


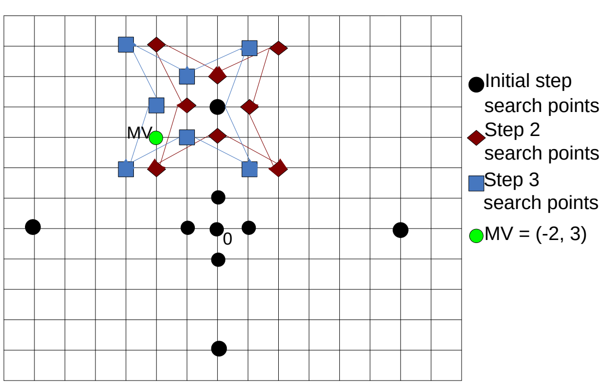
Fig.9 System Diagram of ADLISP-A

Journal of Engineering Technology Vol. 3(1): 18-22, 2013
ISSN 2231-8798

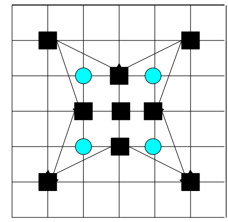
The initial search pattern consists of four inner layer points and this point is then checked to determine the minimum SAD in the center of search window. By condition the minimum SAD still falls in the search center, the minimum SAD estimated the MV and the search is completed. Otherwise, the outer layer which is consisting with four points is then checked. By condition the minimum SAD falls on the inner layer SDSP, the search continues with normal iterative SDSP until found the best possible match. Otherwise if the minimum SAD falls on the outer layer, the following search is performed with several processes.

Basically the previous search is similar to the ADLISP steps. This proposed ADLISP-A will not considered with the adjustment part because of STAR pattern to find the best possible match efficiently. The initial step of the outer layer search points are (±rx, 0), and (0, ±ry) and which form an asymmetric LCSP. By condition the minimum SAD falls on this layer, the search continues with a STAR search on later steps. The search continues with normal iterative STAR until found the best possible match. Iterative STAR process is similar to iterative SDSP except its check pattern is eight-point. The process repeats until the best-matched search point no longer changes.[20,21] The example of this search is shown in Fig.10(a).

Since the eight-side-based fast inner search is used in STAR pattern, there are four more search points are used after applying STAR pattern to determine the final MV. The STAR search pattern as shown in Fig.10(b). The previous ADLISP [12] adopted an E-HEXBS [13] pattern at the asymmetric LCSP (±rx, 0) and (0, ±ry) and achieved slightly poor in PSNR performance compared to ES. Therefore, this paper proposed STAR pattern to replace the E-HEXBS pattern to achieve better PSNR performance.



(a)



(b)

Fig.10 ADLISP-A Description
(a) 8-side fast inner search process when initial search result falls on outer layer
(b) The 4 points of final refinement of STAR pattern.

B. Predicted Motion Vector (PMV) Determination

Motion in most video conference image sequences involves a few blocks and lasts for a few frames. Therefore spatially or temporally adjacent blocks often have similar MVs [14], both spatially and temporally coded MVs representing neighboring MBs are used to predict the MV. As a result to determine the search center before applying the search strategy, the median value between three MVs of spatially adjacent blocks MB0, MB1, and MB2 were considered as illustrated in Fig.11. The candidate MV that yields to the minimum SAD is chosen to be the PMV.

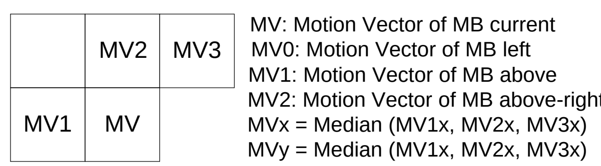


Fig.11: PMV determination approach [15]

Conceptually, the prediction parameters defined in the MPEG-AVC/H.264 standard can be thought of as describing the motion of objects within a sequence of video frames from one frame to the next. The process by which the best prediction parameters are determined is called Motion Estimation [16].

3.0 SIMULATION RESULTS AND DISCUSSION FOR SEARCH PATTERN

To demonstrate the performance of the proposed algorithm, simulations are done to evaluate the computational complexity and PSNR performance. The simulations are performed on two representative .avi video sequences by QCIF frame format (176 x 144) which is Akiyo and Coastguard with 300 total numbers of frames. The proposed ADLISP-A is compared with ADLISP, ARPS, DS, ES, NTSS [17], SESTSS [18], SS4 [19], and TSS. The simulation results are shown in Table I and Table II.

The simulation settings are: 50 frames in which from frame number 151 to 201, 16 x 16 block size, ±16 search window, SAD block distortion measure, and the number of block type, P = 7. The results are tabulated by two testing criteria which are average PSNR per frame and average number of search points per block.

A. Experimental Results for Computational Complexity

Table I
Average Number of Search Points per MB with Different Algorithms and Different Video Sequences

	Akiyo	Coastguard
ADLISP-A	4.578235	10.530200
ADLISP	4.562549	7.255882
ARPS	4.858006	6.904549
DS	11.424790	13.314120
ES	184.555600	184.555600
NTSS	14.666110	16.677190
SESTSS	16.208930	15.812850
SS4	14.657190	15.762520
TSS	21.484800	21.484800

For block-matching, the computational complexity of ME is mainly depending on the average number of search points required for each MB. Based on the simulation results in Table I, the number of search point by the ADLISP is much smaller than other algorithms along with followed by ADLISP-A and ARPS when experimented with Akiyo video sequence. In Coastguard video sequence, ARPS takes up less number of search points amongst all. ADLISP-A is slightly increased by 3.625651 points than ARPS.

B. Experimental Results for PSNR Performance

Table II
Average PSNR per Frame with Different Algorithms and Different Video Sequences

	Akiyo	Coastguard
ADLISP-A	46.50421	32.07022
ADLISP	46.50421	32.06890
ARPS	46.50421	32.07022
DS	46.50421	32.07022
ES	46.50421	32.07022
NTSS	46.50421	32.07022
SESTSS	46.50421	32.03459
SS4	46.50421	32.07022
TSS	46.50421	32.03346

Based on the result in Table II, the proposed ADLISP-A approach pretty close to the PSNR results of the ES for both video sequences. While ADLISP is slightly dropped by 0.00132dB compared with ADLISP-A and ES. It is found that the ADLISP-A used more number of search point but still keeps good quality.

4.0 CONCLUSION

This paper proposes a novel ADLISP-A for fast ME. By using an adaptive size for the initial cross pattern, both large and small motion can be easily estimated.

In support of the best algorithm, the end result should see a PSNR value close to that of ES while computational expense should be significantly less. In comparison of computation cost, ADLISP-A, ADLISP, and ARPS spent least overall search points. It shows that the proposed algorithm is a cost-effective as doing little computation at each possible location in the search window.

On the close observation, ADLISP-A gave significantly better PSNR in which spot a PSNR value close to that of ES. Of the various algorithms simulated during this project experiment, ADLISP-A turns out to be the best fast BMA.

REFERENCES

ITU-T Standard

[1] ITU-T Rec. H.264 | ISO/IEC 14496-10. Information technology — Coding of audio-visual objects, Part 10: Advanced Video Coding, 2002.

[2] ITU-T Recommendation H.263, "Video Coding for Low Bit Rate Communication," Feb. 1998.

[3] ISO/IEC 11172-2, "Information Technology — Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to about 1.5 mbit/s - Part 2: Video," 1993.

[4] ISO/IEC 13818-2 and ITU-T Recommendation H.262, "Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Video," 1996.

Journal and Conference Proceeding

[5] Y.S. Chen, Y.P. Hung, and C.S. Fuh, "Fast Block Matching Algorithm Based on the Winner-Update Strategy", IEEE Trans. on Image Processing, Vol. 10, No. 8, pp. 1212-1222, Aug. 2001.

[6] I Richardson, "Full Search Motion Estimation," in Video CODEC Design, pp. 99-101, 2002.

[7] R. Li, B. Zeng, and M. L. Liou, "A New Three Step Search Algorithm for Block Motion Estimation", IEEE Trans. Circuits System Video Tech., vol. 4, pp. 438–442, 1994.

[8] 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, 2000.

[9] C.-H. Cheung, and L. -M. Po, "Novel Cross Diamond-Hexagonal Search Algorithms for Fast Block Motion Estimation", IEEE Trans. Multimedia, vol. 7, no. 1, pp. 16–22, Feb. 2005.

[10] C. Zhu, X. Lin, and L.P. Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation", IEEE Trans. Circuit and Systems for Video Tech., vol. 12, pp. 349–355, 2002.

[11] Yao Nie, and Kai-Kuang Ma, "Adaptive Rood Pattern Search for Fast Block-Matching Motion Estimation", IEEE Trans. Image Processing, vol.11, no.12, pp. 1442-1448, Dec. 2002.

[12] S. C. Tai, C. S. Yu, and C. W. Lee, "Adaptive Double-Layered Initial Search Pattern for Fast Motion Estimation", IEEE Trans. on Multimedia, vol. 8, no. 6, Dec. 2006.

[13] C. Zhu, X. Lin, L. Chau, and L.-M. Po, "Enhanced Hexagonal Search for Fast Block Motion Estimation", IEEE Trans. Circuits Syst. Video Tech., vol. 14, no. 10, pp. 1210–1214, Oct. 2004.

[14] L. -J. Luo, C. Zou, and X. -Q. Gao, "A New Prediction Search Algorithm for Block Motion Estimation in Video Coding", IEEE Trans. Consumer Electron., vol. 43, pp. 56–61, Feb. 1997.

[15] H.Chaouch, I.Werda, A.Samet, M.A. Ben Ayed, and N. Massmoudi, "Optimal DSP-Based Motion Estimation Tools Implementation For H.264/AVC Baseline Encoder", IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.5, May 2007.

[16] H.Chaouch, I.Werda, A.Samet and N. Massmoudi, "Search Window Impact on H.264/AVC Motion Search Implementation on TMS320C6416 DSP", IEEE Conference on Commun. & Signal Processing. SSD Vol. 3: pp.19-22, March 2007.

[17] Renxiang Li, Bing Zeng, and Ming L. Liou, "A New Three-Step Search Algorithm for Block Motion Estimation", IEEE Trans. Circuits and Sys. for Video Tech., vol 4., no. 4, pp. 438-442, Aug. 1994.

[18] Jianhua Lu, and Ming L. Liou, "A Simple and Efficent Search Algorithm for Block-Matching Motion Estimation", IEEE Trans. Circuits and Sys. for Video Tech., vol 7, no. 2, pp. 429-433, Apr. 1997.

[19] Lai-Man Po, and Wing-Chung Ma, "A Novel Four-Step Search Algorithm for Fast Block Motion Estimation", IEEE Trans. Circuits and Sys. for Video Tech., vol 6, no. 3, pp. 313-317, June 1996.

[20] N.H.M.Yunus and A.H.A. Rahman, "A new Techniques of Block Matching Algorithm for Motion Estimation", in *Proc. 2008 National Seminar on Engineering & Technopreneurship* , pp. 34-38.

[21] N.H.M.Yunus. "Alterated Fast Block Matching Algorithm Based on Adaptive Doubled Layered Initial Search Pattern", Proc. 2010 National Conference Engineering Technopreneurship, pp. 29-32.