The Nucleus 55, No. 4 (2018) 164-170

www.thenucleuspak.org.pk

The Nucleus

ISSN 0029-5698 (Print)
ISSN 2306-6539 (Online)

Modified Adaptive Predict Hexagon Based Search Motion Estimation Algorithm

I. Ali[1], M. Muzammil[2], A. Basit[3*] and I. Haq[2]

[1]National Centre for Physics, Islamabad, Pakistan

[2]International Islamic University, Islamabad, Pakistan

[3]Pakistan Institute of Nuclear Science and Technology, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received : 09 July, 2018
Accepted : 20 November, 2018
Published : 05 December, 2018

Keywords:
Block matching,
Hexagon based search,
HD video,
Motion vector,
Motion estimation

ABSTRACT

Motion Estimation (ME) is a very critical part of any video encoder and different fast ME algorithms are proposed to reduce the overhead of computational cost that was available in encoding system. This paper proposes a Modified Adaptive Predict Hexagon Based Search (MAPHS) algorithm to cope the computational requirements. The proposed algorithm predicts the direction of motion by using adaptive rood shaped predictor and then Hexagonal based Search is applied to refine the search process. The adaptive predictor changes own size predetermined motion vector for each macro block. The ME process is speedup by incorporating thresholding technique, which is primarily beneficial for the videos having low motion activities. Moreover, the proposed algorithm has improved the peak signal to noise ratio (PSNR) by adaptively changing the threshold value according to the motion contents of different videos. The experimental results show that the proposed algorithm performs well on both high definition and common intermediate format videos with respect to ME time and number of search points with acceptable PSNR.

1. Introduction

The block matching Motion Estimation (ME) algorithms have been widely used in current video coding standards like MPEG-1–MPEG-4, H.261–H.264 and HEVC/H.265 to reduce the strong temporal redundancy between successive frames [1–5]. These frames are known as current frame and reference frame. In any ME algorithm, current frame is divided into rectangular, non-overlapping and equivalent size blocks which are called Macro Blocks (MBs). A same MB from a reference frame (may be past or future frame) is found in specific search range by applying a block matching technique. Difference between current MB and reference MB is found by moving MB of reference frame within a search window around the MB of current frame. As the displacement vector of current MB with respect to the best matched reference MB has x and y co-ordinates, so it is considered as a Motion Vector (MV). The displacement vector that has minimum Sum of Absolute Difference (SAD) is considered as best MV which is assigned to the searched MB as a best matching block [6–7].

The ME algorithms which uses the MB based ME approach are called Block Matching ME Algorithms (BMAs). The simplest type of BMA is the Full Search (FS) algorithm which exhaustively searches each possible point in the search window for the best matching block and compute minimum possible SAD. FS may consume up to 80% computational power of the video encoder because of its high computational complexity, so it is not appropriated for real time video coding applications [8–9]. Therefore, different fast ME algorithms are proposed like New Three Step Search (NTSS) [10], Simple and Efficient Search (SES) [11], Four Step Search (FSS) [12], Diamond Search (DS) [13], Hexagon Search (HexS) [14, 15], Adaptive Rood Pattern Search (ARPS) [16], Predict Hexagon Search (PSH) [17] and Flatted Hexagon Search (FHS) [18], etc. The TSS, NTSS, SES and FSS algorithms use different sizes of square shaped search patterns while the DS algorithm adopts a diamond shaped search pattern and provide the faster processing as well as good performance as compare to TSS, NTSS, SES and FSS [19–23]. These algorithms are more attractive for implementation due to their regularity and simplicity. Moreover, the DS algorithm was accepted in the MPEG-4 Video Verification Model, version 14.0 [24].

In this paper simple and fast BMA named algorithm is proposed, which performs better than major existing algorithms according to number of search points and Speed Improvement Rate (SIR) with approximately same PSNR. The algorithm finds targeted MV by utilizing three kinds of fixed search patterns and one adaptive rood shape pattern. Speed up the search procedure is also integrated by early termination scheme in the proposed algorithm, which is especially advantageous for those video sequences having small motion activities. We tested the proposed algorithm by using CIF (Common Intermediate Format) and High Definition (HD) video sequences. The simulation results show that the proposed algorithm performs well in term of ME time and number of search points, with satisfactory PSNR. Moreover, the PSNR is also improved by reducing threshold value than other considered algorithm.

Remaining paper is organized as follows: Section 2 describes the relationship between the ME accuracy and the number of search points. Proposed algorithm is explained in section 3, results are presented in section 4 and finally conclusion is drawn in Section 5.

* Corresponding author : abdulbasit1975@gmail.com

2. Motion Estimation Accuracy and Search Points Relationship

The search speed and accuracy of ME process are affected by size and shape of search pattern [17, 25–26] which depend upon number of search points. The large search patterns have more number of search points and produce good result in term of accuracy, but may produce delay in search time, particularly for those video sequences which contain small motion content like reading news and video conferencing. Whereas, small search pattern reduce the ME time, but the search may trapped in local minima of search range particularly for those scenarios having high motion activity. Therefore, it should be pointed that the quantity of search point in each iteration of ME process has strong relation with quality and estimation time.

Different famous search patterns like diamond, hexagon, flatted hexagon and one step square which are used in ME algorithms are shown in Fig. 1.

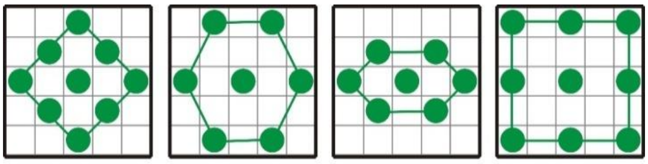

Fig. 1: Search patterns: (a) Diamond, (b) Hexagon, (c) Flatted hexagon, (d) One step square.

The pattern used in DS algorithm performs better than square search pattern which is used in FSS or TSS due to its compact shape and medium size. It can find large MVs with nine search points to maximize the search quality and minimize the ME time. The local minimum trapping is minimized by unrestricted search process in DS algorithm. Hexagon based search pattern with seven points is proposed to reduced search time than DS but it also causes the reduction of MV accuracy in terms of PSNR. Therefore, a flatted hexagon search pattern is proposed which is similar to diamond search pattern except reduction of two top and bottom points. The flatted hexagon provides better results in terms of MV accuracy than hexagon pattern with minimum ME time.

The shape of search pattern with different number of search points, affects on ME time and accuracy. Moreover, the selection of search points is based on probability distribution of motions in any video frame, for example horizontally biased motion or vertically. The motion distribution probability is found by computing angles of MVs. A motion considered as horizontally biased motion if the angle of any MV is in the range of 330º– 30º and 150º–210º; whereas, motion is vertically biased if the angle of MV is in the range of 60º–120º and 240º– 300º. The probability of motion distribution for different video sequences is given by Chen and Li [18] which shows that the sequence football and garden have 17.05% and 12.12% vertical bias motion; whereas, 27.14% and 92.78% horizontal motion, respectively. This shows that the search pattern should be horizontal or vertical for finding the best MV. Therefore, the speed and performance in ME process may be improved by implication of horizontal and vertical flatted search patterns. In addition, incorporation of adaptive prediction with horizontal and vertical flatted search patterns can speed up the ME process.

3. Proposed Algorithm

The proposed "Modified Adaptive Prediction based Hexagon Search" (MAPHS) algorithm computes the best MVs by employing the fact of coherency in any frame, which states that the previously calculated MV is mostly same as MVs of adjacent MBs [16]. Moreover, it also considers the hypothesis of uni-modal surface which expresses that the SAD magnitude decreases monotonically as the search of best MV moves toward the global minimum error location. The algorithm applies four different types of search patterns

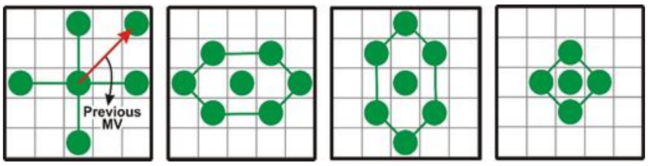

Fig. 2: Search patterns used to find the MV.

to find the MV, i.e., Adaptive Rood Shaped Predictor (ARSP), Horizontal Flatted Hexagon Search Pattern (HFHSP), Vertical Flatted Hexagon Search Pattern (VFHSP) and Unit Step Search Pattern (USSP) as shown in Fig. 2.

The ARSP is used for fast prediction of motion direction which leads to selection of HFHSP or VHSP, which are used for refinement search. The USSP is used for final selection of best MV. Early termination procedure is also combined in the proposed algorithm by a threshold value to speed up the search, which is especially helpful for those videos having small motion activities. The optimum value of threshold is selected by identifying total number of static MB in different video sequences, using FS ME algorithm. It is observed that many video sequences like video conferencing have high percentage of static MBs between consecutive frames. Table 1 shows the average static MBs per frame for different videos [16].

From Table 1 it is observed that except Foreman, all video sequences have more than 70% static MBs in a frame. Therefore, it is possible to reduce ME time significantly if early termination scheme is applied at the start of ME process. It is examined that the average SAD of these static blocks is very low as compared to moving blocks. Consequently, SAD is computed from initial location of current and reference MBs to employ early termination. After this, SAD is compared with predefined threshold (T). If SAD is less than T, then the current MB is declared static by assigning (0, 0) MV and further search is stopped. Average value of SAD for static MBs varies from 600 to 1300. These results were obtained by experimenting FS on different video sequences. A threshold value of

T= 512 is selected in the proposed scheme, as the minimum SAD value for the static MBs is still higher than 512. This value can be adaptively adjusted according to motion activity in a video sequence.

Table 1: Average percentage of static macro blocks per frame.

Video sequence	Coding bit-rate (Kbps)	Avg. static MBs % per frame
AKIYO	10	93.36%
CONTAINER	10	90.55%
HALL)	10	95.18%
NEWS	112	84.61%
MOTHER DAUGHTER	24	80.36%
SILENT	24	78.79%
TENNIS	1024	70.44%
FOREMAN	1024	37.41%

The proposed algorithm is described in four steps as follows:

Step 1: Compute the value of SAD at center of ARSP and compare with threshold value. If SAD is less than this threshold value then this particular MB is considered static MB and the search is terminated by assigning (0, 0) to MV; otherwise, go to Step 2 for further search. In videos with low motion activity, the threshold is set to lesser value and vice versa, which speed up the ME process.

Step 2: Direction of motion is predicted by searching the remaining four points of ARSP. Moreover, search the location of previous MV as sixth point of ARSP. Size of ARSP is determined with equation (1).

$$Step\_Size = Max\left(\left|MV_x\right|, \left|MV_y\right|\right) \quad (1)$$

Where, MV_x correspond to x-coordinate and MV_y correspond to y-coordinate of the previously calculated MV. Step size of ARSP is one, if MV of previous MB is (0, 0). It is also one, when the search is carried out for the MB which is located at the first column of a video frame. After that apply search to five points of ARSP and if step size is equal to one and minimum SAD is at center point, assign (0, 0) to MV and terminate the search (Early Termination), otherwise go to Step 3.

Step 3: Predict the direction of motion horizontal or vertical and then set the minimum SAD point as origin of new search pattern. Search all points of hexagon search pattern and continue the search process until minimum SAD is obtained at the center point then proceed to Step 4. HFHSP is applied when predicted motion direction is horizontal, whereas, in other case search process is continued by using VFHSP.

Step 4: Use USSP to get the best matching point and assign the MV to this particular MB.

There are some conditions when MV is assigned. For example, when threshold criteria is matched at (0, 0) location; similarly, at step one when S = 1 and minimum SAD is at center of ARSP and when USSP is applied in fourth step of proposed algorithm. The hexagon based search pattern is applied in all the cases where minimum value of SAD not exists at center of ARSP and hexagon, while the USSP is applied only in two cases; first, when the minimum SAD is at the center of ARSP; the second, when minimum SAD is at the center of hexagon.

4. Experimental Results

The proposed algorithm is applied to a number of publicly available video sequences and the results are mentioned in this section. Details of datasets and performance metrics are also described.

4.1. Simulation Parameters and Used Dataset

The compilation and simulation of proposed algorithm is achieved by using MATLAB 2014a. The system which is used for complete implementation has following specifications: Intel® Core™ 2 Duo CPU T6570 @ 2.10G processor, DDR3 RAM (4 GB), video card of 512 MB memory, 2 MB L2 Cache, 800 MT/s FSB and 64-bit Windows® 7 operating system. Various simulation parameters were used as given in Table 2.

Table 2: Parameters for simulation.

Parameters for simulation	Value
Number of frames	100
Image size	CIF (352×288)
Video format	YUV
YUV Subsampling format	4:2:0
Search range/parameter (p)	7
MB Size	16×16
Threshold (T)	0, 512

The dataset which is used for performance analysis of proposed algorithm comprise different types of CIF and HD video sequences. All sequences have different motion activity (low, moderate and high) and different direction of motion as given in Table 3 [27].

All the video sequences are in YUV format with 4:2:0 chroma sub-sampling mode with frame size of 720p for HD and 352×288 for CIF [28]. The 100 frames of each video sequence are utilized to acquire simulation results. Four performance parameters namely: Mean Square Error (MSE), PSNR, Search Points and ME time are considered to calculate the performance of the proposed algorithm. The results of proposed algorithm for above parameters are compared with major existing algorithms like DS, HexBS, FHS and FSS.

Table 3: Different types of video sequences according to motion activity.

Video Sequence	Classification of motion activity	Motion direction
Football	High	Multidirectional
Soccer	High	Multidirectional
Stefan	High	Multidirectional
Mobile	High	Multidirectional
Flower	High	Multidirectional
Coastguard	High	Unidirectional
Paris	Moderate	Multidirectional
Foreman	High	Multidirectional
Car phone	Moderate	Multidirectional
Hall	Low	Unidirectional
Silent	Moderate	Multidirectional
News	Moderate	Unidirectional
Container	Low	Unidirectional
Bridge Close	Low	Unidirectional
Bridge Far	Low	Unidirectional
Mother Daughter	Moderate	Unidirectional
Miss America	Low	Unidirectional
Akiyo	Low	Unidirectional
Highway	High	Unidirectional
Don’t Look	High	Unidirectional
Old Town Cross	Moderate	Multidirectional
Into Tree	Moderate	Multidirectional

4.2. Mean Square Error Analysis

MSE is calculated by taking the mean of SAD which are mathematically represented in equations (2) and (3):

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \qquad (2)$$

$$MSE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(C_{ij} - R_{ij}\right)^2 \qquad (3)$$

Where N^2 is the total numbers of pixels of a MB while C_{ij} and R_{ij} are the intensity values at the pixel (i, j) of current MB and reference MB, respectively. If value of MSE is near to zero then the predicted MV will be more accurate. The result of average MSE of 100 frames of different video sequences for the proposed algorithm is shown in Fig. 3. We observe that the proposed algorithm perform equally as compared with other algorithms. Moreover, it is better (in term of MSE) than HexS and PHS for some cases like “Football”, “Stefan”, “Coastguard” and “Don’t Look”, etc.

4.3. Peak Signal to Noise Ratio Analysis

The PSNR might be considered as function of MSE and calculated by using the mathematical relation which is given in equation (4):

$$PSNR = 10Log_{10}\left[\frac{(255)^2}{MSE}\right] \qquad (4)$$

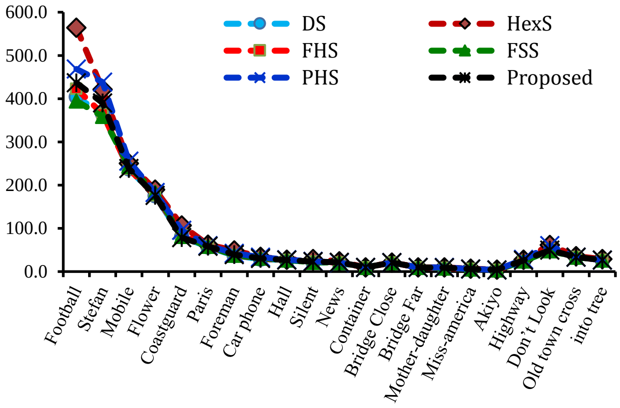

Fig. 3: Average MSE of 100 frames for different videos.

PSNR demonstrates about the motion in compensated image that is created by using MVs of MBs from the reference frame. Additionally, PSNR describes the quality of regenerated video by using computed MVs. The average PSNR value of 100 frames from different video sequences is calculated to compare the quality of regenerated video by using the MVs which are computed through proposed algorithm. The results for average PSNR (see Table 4) show that the proposed algorithm performs better than HexS, PHS and FSS algorithms.

Moreover, the PSNR of the proposed algorithm was improved by reducing the value of threshold from 512 to 0. The PSNR improved 2.8% than HexS for Foreman, 0.8% and 0.1% than FSS and FHS, respectively; for Flower 2.1% than PHS for Coastguard and 0.2% than DS for Miss America, when threshold kept zero. Furthermore, the average search points is also increased for proposed algorithm by decreasing the value of threshold which causes to increase ME time but it remains less than all other algorithms for this scenario.

4.4. Number of Search Point Analysis

Number of search points is an important factor for the development of any algorithm because it directly affects the search speed and computational complexity of the algorithm. Hence, the number of search points for DS and HexBS algorithms can be calculated using equations (5) and (6), respectively [15].

$$N_{DS} = 9 + (M \times n) \qquad (5)$$

$$N_{HexS} = 7 + (3 \times n) + 4 \qquad (6)$$

Where “n” is number of steps in second step of HexBS and DS algorithms while “M” may be 3 or 5 search points, which depends on search direction. The average number of search points per MB for hundred frames from different video sequences are calculated and given in Table 5.

Table 4: Average PSNR of 100 video frames.

Video sequence	DS	HexS	FHS	FSS	PHS	MAPHS
Football	22.80	21.37	22.59	22.91	22.15	22.50
Soccer	24.01	23.55	24.01	24.07	23.21	23.66
Stefan	24.39	24.18	24.39	24.28	24.08	24.38
Mobile	25.88	25.53	25.86	25.68	25.70	25.87
Flower	29.81	29.46	29.74	29.75	29.14	29.76
Coastguard	30.63	30.27	30.55	30.54	30.44	30.53
Paris	32.77	31.66	32.73	32.65	32.28	32.52
Foreman	33.65	32.96	33.56	33.36	33.16	33.51
Car phone	34.26	34.15	34.22	34.26	34.14	34.18
Hall	35.61	34.71	35.47	35.66	35.28	35.42
Silent	37.12	36.86	37.06	37.07	37.00	37.00
News	38.33	38.33	38.33	38.33	38.33	38.33
Container	35.01	35.01	35.01	35.01	35.01	35.01
Bridge Close	38.39	38.39	38.39	38.39	38.38	38.37
Bridge Far	39.90	39.50	39.83	39.91	39.61	39.78
Mother Daughter	40.43	40.04	40.45	40.31	40.30	40.27
Miss America	42.82	42.56	42.80	42.78	42.80	42.79
Akiyo	34.49	33.83	34.47	34.38	33.73	34.07
Highway	31.99	30.81	31.82	32.11	30.81	31.96
Don't Look	33.00	32.71	32.99	32.94	32.94	32.97
Old Town Cross	33.93	33.57	33.92	33.84	33.86	33.88
Into Tree	22.80	21.37	22.59	22.91	22.15	22.50

It can be observed from Table 5 that the proposed algorithm takes less number of search points than all considered algorithms which causes to speed up the ME process and reduce the ME time. Therefore, the proposed algorithm has the ability to cope the requirements of real time implementation.

4.5. Motion Estimation Time Analysis

The ME time for each frame of each video sequence is computed and averaged by hundred for each scenario, as shown in Fig. 4. Table 6 shows the detailed results for ME time. By keeping in mind the result for number of search points we can observed that the propose algorithm is also fast in term of ME time. It computes best MVs in minimum time for all HD and CIF video sequences.

Table 5: Average number of search points of 100 video frames.

Video sequence	DS	HexS	FHS	FSS	PHS	MAPHS
Football	21.56	12.16	17.93	22.08	16.49	14.18
Soccer	16.31	12.56	13.51	18.74	11.01	9.24
Stefan	13.47	10.59	11.50	16.35	7.19	6.81
Mobile	15.30	11.88	12.80	18.41	9.96	6.29
Flower	16.04	12.59	13.54	18.70	11.30	9.55
Coastguard	12.91	10.56	10.90	16.31	6.06	3.22
Paris	15.59	11.23	13.23	18.25	10.32	7.73
Foreman	14.85	11.03	12.44	17.81	9.03	6.48
Car phone	12.87	10.42	10.67	16.31	5.82	3.84
Hall	13.04	10.45	11.06	16.37	6.00	3.03
Silent	12.57	10.43	10.63	16.05	5.49	2.03
News	12.34	10.41	10.44	15.93	5.01	1.53
Container	13.56	10.59	11.58	16.67	6.25	4.07
Bridge Close	13.05	10.79	11.03	16.42	5.99	3.81
Bridge Far	13.44	10.55	11.28	16.82	6.69	2.59
Mother Daughter	15.47	10.77	12.78	18.20	8.10	3.05
Miss America	12.28	10.32	10.39	15.85	4.96	1.47
Akiyo	15.03	11.43	12.79	17.90	8.91	6.22
Highway	19.91	12.46	16.88	21.33	14.59	6.72
Don't Look	15.07	11.66	12.81	18.69	8.85	7.91
Old Town Cross	15.35	11.91	13.03	18.74	9.83	8.29
Into Tree	21.56	12.16	17.93	22.08	16.49	14.18

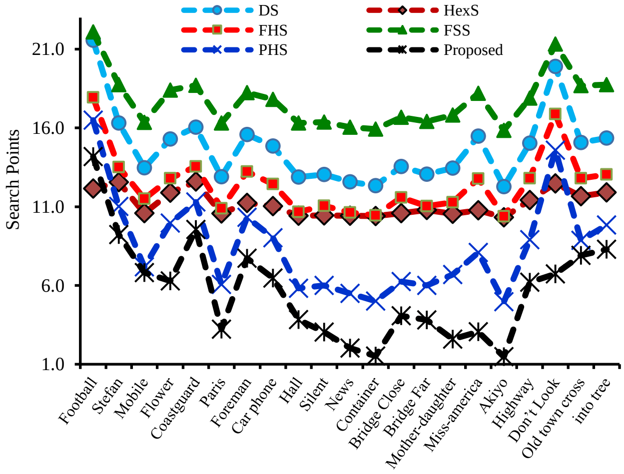

Fig. 4: Average motion estimation time (in seconds) of 100 frames for different videos.

Table 6: Average motion estimation time (sec.) of 100 video frames.

Video sequence	DS	HexS	FHS	FSS	PHS	MAPHS
Football	0.3145	0.1806	0.2458	0.1899	0.3207	0.2316
Soccer	0.2272	0.1776	0.1916	0.1654	0.2076	0.1527
Stefan	0.1846	0.1546	0.1661	0.1394	0.1204	0.1091
Mobile	0.2152	0.1764	0.1859	0.1619	0.1769	0.1044
Flower	0.2079	0.1741	0.1854	0.1566	0.1912	0.1476
Coastguard	0.1729	0.1514	0.1549	0.1403	0.0994	0.0518
Paris	0.2143	0.1624	0.1912	0.1568	0.1791	0.1238
Foreman	0.1953	0.1573	0.1721	0.1487	0.1489	0.1004
Car phone	0.1790	0.1566	0.1547	0.1399	0.0955	0.0600
Hall	0.1794	0.1539	0.1644	0.1421	0.0987	0.0474
Silent	0.1712	0.1545	0.1562	0.1417	0.0913	0.0317
News	0.1670	0.1515	0.1513	0.1372	0.0805	0.0206
Container	0.1910	0.1508	0.1623	0.1393	0.1072	0.0562
Bridge close	0.1738	0.1525	0.1573	0.1379	0.0955	0.0581
Bridge far	0.1834	0.1585	0.1669	0.1437	0.1102	0.0416
Mother Daughter	0.2042	0.1558	0.1802	0.1582	0.1384	0.0479
Miss America	0.1643	0.1487	0.1499	0.1314	0.0783	0.0236
Akiyo	0.2086	0.1769	0.1825	0.1591	0.1539	0.1013
Highway	2.5184	1.6196	2.0723	1.6466	2.3963	0.9587
Don’t look	2.0259	1.6785	1.8102	1.6127	1.4524	1.2291
Old town cross	1.8835	1.5627	1.6919	1.4615	1.5318	1.2013
Into tree	0.3145	0.1806	0.2458	0.1899	0.3207	0.2316

5. Conclusions

In this paper a fast block matching Modified Adaptive Predict Hexagon Based Search (MAPHS) ME Algorithm is proposed which speeds up the ME process significantly with no degradation of MSE and PSNR. The proposed MAPHS is tested on HD and CIF video sequences with multidirectional high motion activity. In all cases the proposed algorithm utilizes less number of search points and requires minimum ME time to compute the best MV. Moreover, it out performs HexS, PHS and FSS in terms of all parameters including MSE and PSNR for the video sequence having high and multi direction motion activity. The PSNR is also improved by reducing the threshold value to zero with minimum ME time and search points for some cases. As the proposed algorithm consumes minimum ME time for any type of video sequence. Therefore, it can be used for real time video applications like HD videos in H.264/AVC or MPEG-4.

References

[1] K.R. Rao and J.J. Hwang, “Techniques and standards for image, video and audio coding”, Prentice Hall PTR, 1996.

[2] K. Rijkse, "H.263: video coding for low-bit-rate communication", Communications Magazine, IEEE, vol. 34, no.12, pp. 42-45, Dec 1996.

[3] T. Wiegand, “Draft ITU-T recommendation and final draft international standard of joint video specification”, Joint Video Team JVT-G050 Pattaya, Thailand, 2003.

[4] G.J. Sullivan, J. Ohm, Woo-Jin Han and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard", IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 12, pp. 1649 -1668, 2012.

[5] R. Takada, S. Orihashi, Y. Matsuo and J. Katto, "Improvement of 8K UHDTV picture quality for H.265/HEVC by global zoom estimation", IEEE International Conference on Consumer Electronics, pp. 58-59, Jan. 2015.

[6] I. Ali, M. Muzammil and G. Raja, “Performance analysis of motion estimation algorithms based on motion activity in video sequences”, Pakistan Journal of Science, vol. 64, pp. 39-45, March, 2012.

[7] S.D. Kamble, S.T. Khawase, N.V. Thakur and A.V. Patharkar, "An improved approach of block matching algorithm for motion vector estimation", Int. J. Inf. Retrieval Res. vol. 8, no. 1, 38-56 2018.

[8] Y. Li, G. Yang, Y. Zhu, X. Ding and X. Sun, "Unimodal stopping model-based early SKIP mode decision for high-Efficiency video coding", IEEE Transactions on Multimedia, vol. 19, no. 7, pp. 1431-1441, 2017.

[9] P. Nguyen, H. Tran, H. Nguyen, X-N. Nguyen, C. Vo, B. Nguyen, V-D. Ngo and V-T. Nguyen, "Asymmetric diamond search pattern for motion estimation in HEVC", IEEE Fifth Int. Conf. on Communications and Electronics, pp. 434-439, July 30, August 1, 2014.

[10] L. Renxiang, B. Zeng and M.L. Liou, "A new three-step search algorithm for block motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, no. 4, 1994.

[11] L. Jianhua and M.L. Liou, "A simple and efficient search algorithm for block-matching motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 7, no. 2, 1997, pp. 429-433.

[12] L-M. Po and W-C. Ma, "A novel four-step search algorithm for fast block motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 6, no. 3, pp. 313-317, June 1996.

[13] Z. Pan, R. Zhang, W. Ku and Y. Wang, "Adaptive pattern selection strategy for diamond search algorithm in fast motion estimation", Multimedia Tools and Applications, 1-18, 2018. https://doi.org/ 10.1007/s11042-018-6353-2.

[14] C. Zhu, X. Lin, L-P. Chau, K-P. Lim, H-A. Ang and C-Y. Ong, "A novel hexagon-based search algorithm for fast block motion estimation", IEEE Conference on Acoustics, Speech and Signal Processing, vol. 3, pp. 1593-1596, 2001.

[15] C. Zhu, X. Lin and L-P. Chau, "Hexagon-based search pattern for fast block motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no. 5, pp. 340-355, 2002.

[16] S. Yavuz, A. Celebi, M. Aslam and O. Urhan, "Selective gray-coded bit-plane based low-complexity motion estimation and its hardware architecture", IEEE Transactions on Consumer Electronics, vol. 62, no. 1, pp.76-84, 2016.

[17] T-H. Tsai and Y-N. Pan, "A novel predict hexagon search algorithm for fast block motion estimation on H.264 video coding", Circuits and Systems, Proc. of IEEE Asia-Pacific Conference, pp. 609-612, vol. 1, 6-9 Dec. 2004.

[18] T-H. Chen and Y-F. Li, "A novel flatted hexagon search pattern for fast block motion estimation", Int. Conf. on Image Processing, vol. 3, pp. 1477-1480, 24-27 Oct. 2004.

[19] F.H. Jamil, A. Chekima, R.R. Porle, O. Ahmad and N. Parimon, "BMA performance of video coding for motion estimation", IEEE Third Int. Conf. on Intelligent Systems, Modelling and Simulation, pp. 287-290, 2012.

[20] Z. Cuanfeng, L. Guizhong and S. Rui, "A modifed hexagon-based search algorithm for block motion estimation", IEEE Int. Conf. Neural Networks & Signal Processing Nanjing. China, pp. 1205-1208, December 14-17, 2003.

[21] D.V. Babu, P. Subramanian and C. Karthikeyan, "Performance Analysis of block matching algorithms for highly scalable video compression", IEEE Conf. Ad Hoc and Ubiquitous Computing, pp. 179-182, 2006.

[22] R.A. Manap, S.S.S. Ranjit, A.A. Basari and B.H. Ahmad, "Performance analysis of hexagon-diamond search algorithm for motion estimation", IEEE Int. Conf. Computer Engineering and Technology, vol. 3, pp. V3-155-V3-159, 2010.

[23] J. Luo, X. Yang and L. Liu, "A fast motion estimation algorithm based on adaptive pattern and search priority", Multimedia Tools and Applications, vol. 74, no. 24, pp. 11821-11836, 2015.

[24] M. Muzammil, G. Raja and I. Ali, "Field programmable gate array (FPGA) architecture of diamond search motion estimation algorithm for real-time video applications", NED University Journal of Research - Applied Sciences, vol. XII, No. 4, 2015.

[25] B. Zou, C. Shi, C-H. Xu and S. Chen, "Enhanced hexagonal-based search using direction-oriented inner search for motion estimation", IEEE Transactions on Circuits and Systems for Video Technology, vol. 20, no. 1, pp.156-160, 2010.

[26] I. Ali, G. Raja, M. Muzammil and A.K. Khan, "Adaptive modified hexagon based search motion estimation algorithm", IEEE 4th Intl. Conf. on Consumer Electronics, Berlin, pp.147-148, 7-10 Sept., 2014.

[27] G. Raja, M.J. Mirza and T. Song, "H.264/AVC de-blocking filter based on motion activity in video sequences", IEICE Electronics Express, vol. 5, no. 19, pp. 809-814, 2008.

[28] CIF format Videos, "YUV test video sequences", available on-line: http://media.xiph.org/video/derf/, Jan-2016.