Modified Cross Hexagon Diamond Search Algorithm for Fast Block Matching Motion Estimation

R. Vani [#1], P. Davis [*2], M. Sangeetha [*3]

[#] ECE Department, Anna University,
Meenakshi College of Engineering, Chennai, India.
[1] vani_gowtham@yahoo.com
[*] ECE Department, Anna University
Karpaga Vinayaga College of Engineering and Technology, Chennai, India.
[2] dpdavis91@gmail.com
[3] sang_gok@yahoo.com

Abstract—The most essential part in Video Coding Standards is Motion Estimation. But, Motion Estimation itself consumes more than half of the coding time to encode. To reduce the computational complexity, the time required for Motion Estimation should be reduced for which efficient algorithms are required. A new Modified Cross Hexagon Diamond Search (MCHDS) algorithm is proposed which will reduce the motion estimation time with no loss in quality compared to the existing algorithms. The proposed MCHDS algorithm reduces an average of 18.05 % of Motion Estimation time compared to Hexagon Search (HS), 35.91% of time for motion estimation compared to Diamond Search (DS) and 33.32% of Motion Estimation time compared to New Hexagon Search (NHEXS). The average number of search points is also reduced with no loss in quality.

Keyword-H.264, Search points, Algorithm, Motion Estimation time, Computations

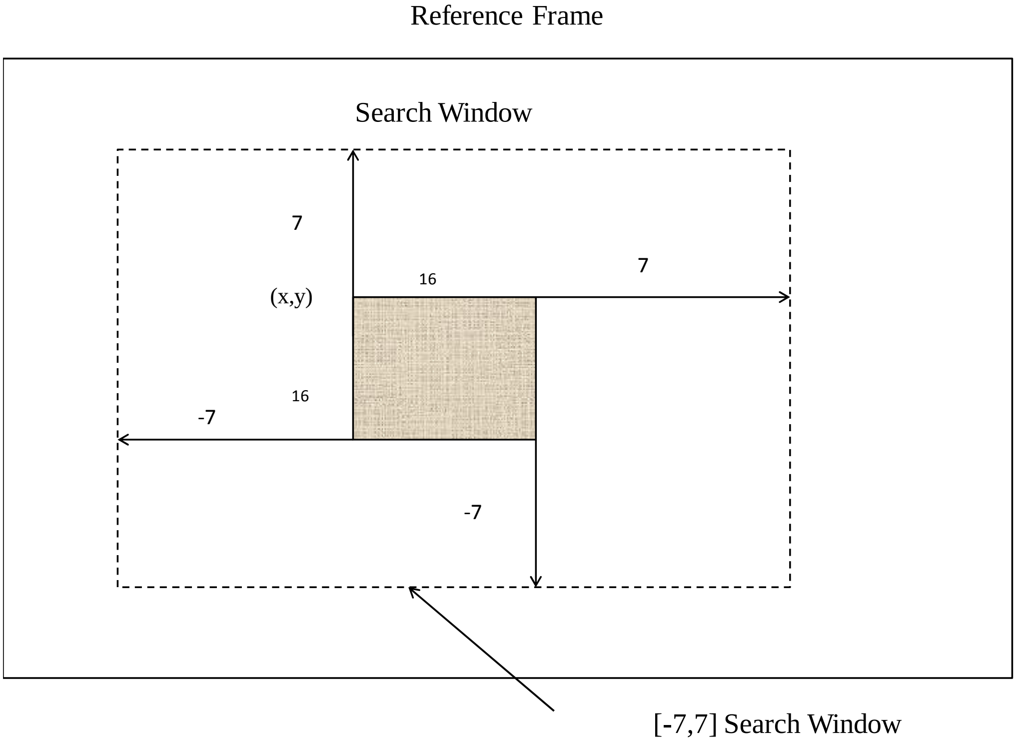
I. Introduction

Video Coding Technology is a research focus of digital video media. Motion Estimation (ME) is the most crucial part of video compression as it consumes about 80% of the total computational complexity of total encoding time in video coders and it is used to identify the motion in a video. It is a process of estimating the pixels of the current frame from a reference frame. ME algorithms are used to compute the displacement between current frame and reference frame as shown in Fig.1a and Fig.1b. The motion estimator has two inputs: a macroblock (MB) from the current frame and a pixel search area from the previous frame. For each MB in the current frame, a search window is defined around a point in the reference frame. A distortion measure is defined to measure the similarity between the candidate MB and the current MB. A search is performed within the search window for the best matched candidate MB with maximum similarity. The displacement of the best matched MB from the current MB is the Motion Vector (MV) [2], [8], [12] as shown in Fig.1c. The similarity criterion is used to compare the candidate blocks with each original block and in our work we use Mean Absolute Difference (MAD) as the criteria given by the mathematical expression in (1)

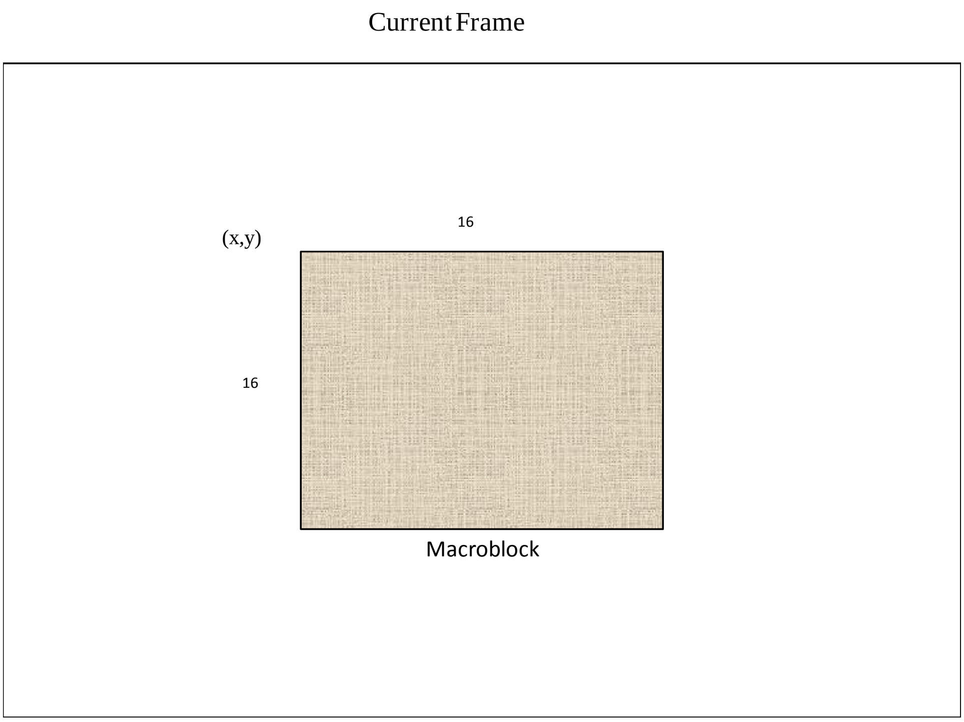
$$MAD(x,y,u,v)=\frac{1}{256}\sum_{i=0}^{i=15}\sum_{j=0}^{j=15}\left|A_{(x+i,y+j)}-B_{((x+u)+i,(y+v)+j}\right| \quad (1)$$

where, (x,y) is the position of current block and (u,v) denotes the motion vector of current block (A) relative to the block in the reference frame (B).

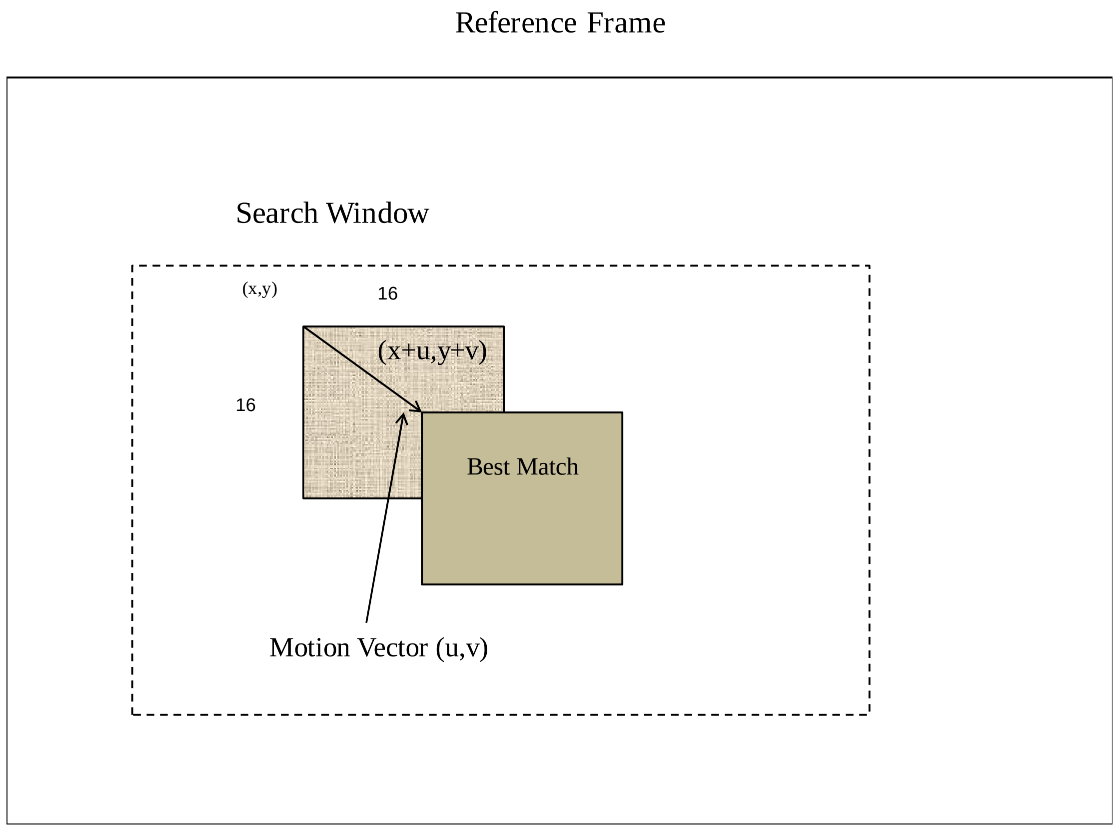
There are two basic approaches to find out the ME, one is called pixel based ME and the other is block based ME. The pixel based ME technique is also known as the optical flow method. It works on the basic assumption of brightness constancy which states that the intensity of a pixel remains constant when it is displaced in the video sequence. Here, MVs are evaluated for every pixel in the frame. In block based ME approach, the current frame is divided into non overlapping Blocks. H.264/AVC is a Video Codec standard for video compression and has a wide range of application from low bit rate (internet streaming) to High Definition Television (HDTV) broadcast. In H.264, the seven possible modes of a block are 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4 as shown in Fig. 2.These are called macro blocks and for each such current frame, one best MV is calculated in the reference frame. Here, an inherent assumption is made that the entire block undergoes translational motion. The algorithms that use block based ME technique are known as block matching ME algorithms. Block based ME algorithms remove temporal redundancy between two or more successive frames and forms an integral part for most of the video coding standards. Block-matching algorithms are used widely due to its simplicity and it is easy to be applied. Since ME process consumes up to 60–80% of computational power of the encoder process, still researches are carried out for an efficient and fast motion estimation algorithms which becomes significant [6].



(a)



(b)



(c)

Fig. 1. (a) Reference Frame with search window (b) Current Frame (c) Best matched Reference block and Motion Vector

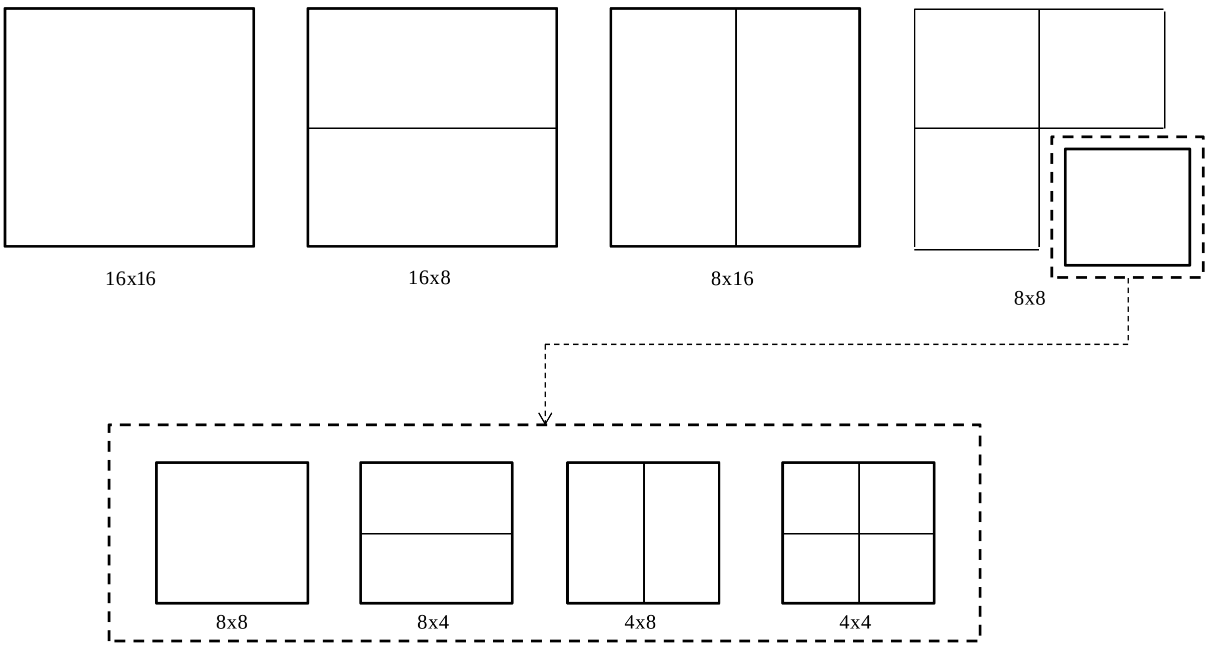


Fig. 2. Variable block size in H.264

The video coding standards like MPEG2, H.264/AVC [20] and the emerging High Efficiency Video Coding Standard (HEVC) [7] do not restrict how the ME is done. Based on this fact, there is a vast space to explore a new algorithm as a solution for the ME. These solutions are evaluated according based on complexity and objective quality of the digital video. In the literature, a number of fast ME algorithms have been proposed. These algorithms include fixed set of search patterns such as Three Step Search (3SS) [17], New Three Step Search (NTSS) [16], Four Step Search (4SS) [18], Diamond Search (DS) [19] and Hexagon Search (HS)[2], [3]. 3SS, NTSS and DS uses rectangular search patterns of different sizes to fit the center biased motion vector distribution characteristics. HS search employs a hexagon shaped pattern and results in fewer search points with similar distortion. The drawback is that these algorithms suffer from the local minima problem and have less adaptability and search efficiency to track large motions. They are based on the assumption that the error surface increases monotonically as the search position moves away from the global minimum. In order to track large motions, advanced algorithms that include adaptive search patterns are used. And hybrid algorithms which combine various search patterns to avoid being trapped in local minima are also used recently. In this paper, a novel fast block matching hybrid algorithm called new Modified Cross Hexagon Diamond Search algorithm (MCHDS) is proposed. It uses cross-shaped search pattern as the initial step and asymmetric hexagon-shaped patterns and small diamond patterns as the subsequent steps for fast block motion estimation (BME). It results in higher motion estimation speed and also reduces the number of search points.

This paper is organized as follows. The motivation of this paper is presented in Section II. Section III explains about the existing Search Patterns. Section IV briefly discusses about the proposed MCHDS search. Section V shows the experimental results in a variety of video sequences to verify the efficiency of the proposed algorithm followed by the concluding remarks in Section VI.

II. Motivation

The motion estimation process comprises basically two steps, namely, (i) the low-resolution coarse search to identify small area where the best motion vector is expected, and (ii) fine-resolution inner search to select the best motion vector in the located small region. The fast algorithms focus on speeding up the coarse search by considering various ways to reduce the number of search points in identifying a small area for inner search. These fast ME algorithms consist of a prediction process and a block matching search process. The former uses one or more predictors such as median predictor, up layer predictor, corresponding block predictor, neighbouring reference frame predictor, (0, 0) etc., to find the initial search point. The predictive way of using the neighbouring motion vectors to find a good starting point can also be employed to reduce the number of search points in the coarse search before the inner search. The block matching process is then used to locate the best matching block by using one or more search patterns [1], [10]. Most of the search algorithms focus on improvement of either coarse search or inner search. Thereby, a fast inner search is desirable and in our opinion, appropriate utilization of the surrounding information in the given search region would yield a faster and accurate inner search. Therefore, the reduction of number of search points is expected to be from two aspects, i.e., one from the prediction for a good starting point, and the other from the proposed fast inner search. This is been adopted in our algorithm to achieve overall minimum number of search points.

III.Existing Search Patterns

In this section, we have described about the existing Search Patterns that are used to find the global minimum points. These patterns come in different forms such as diamond, square or hexagon grids and are used so as to reduce the computational complexity for finding motion vectors. All these algorithms are based on the center biased MV distribution characteristics [14].

A. Diamond Search (DS)

The New diamond search algorithm (DS) has been proposed by S. Zhu and K. K. Ma in 2000. DS algorithm is similar as 4SS, but the search pattern is changed from a square to a diamond, and there is no limit on the number of steps that the algorithm can take. The DS algorithm utilizes two different types of fixed search patterns. The first pattern called large diamond search pattern (LDSP) and the second pattern called small diamond search pattern (SDSP) shown in Fig. 3. The LDSP consisting of nine checking points from which eight points surround the center one to compose a diamond shape. The SDSP consisting of five checking points and it forms a smaller diamond shape [9], [11], [13], [15]. The DS is used as inner search in the final step for obtaining absolute motion vector in our algorithm.

The advantages of DS can be summarized as: in large motion content, optimal minimum can be got using fewer points. It is faster than the traditional method of 4SS for small motion contents. In terms of search mode, it is highly sensitive in all directions. It is harder to fall into the local minimum, when compared to other methods. DS performs well for the image sequence with wide range of motion content. It achieves close MSE performance compared to NTSS while reducing computation by 22% approximately. The diamond shape has high search efficiency and most video sequences contains translation, zooming and tilt motions, that easily fall on the diamond corners when block based ME is employed.

The advantages of DS can be summarized as: in large motion content, optimal minimum can be got using fewer points. It is faster than the traditional method of 4SS for small motion contents. In terms of search mode, it is highly sensitive in all directions. It is harder to fall into the local minimum, when compared to other methods. DS performs well for the image sequence with wide range of motion content. It achieves close MSE performance compared to NTSS while reducing computation by 22% approximately. The diamond shape has high search efficiency and most video sequences contains translation, zooming and tilt motions, that easily fall on the diamond corners when block based ME is employed.

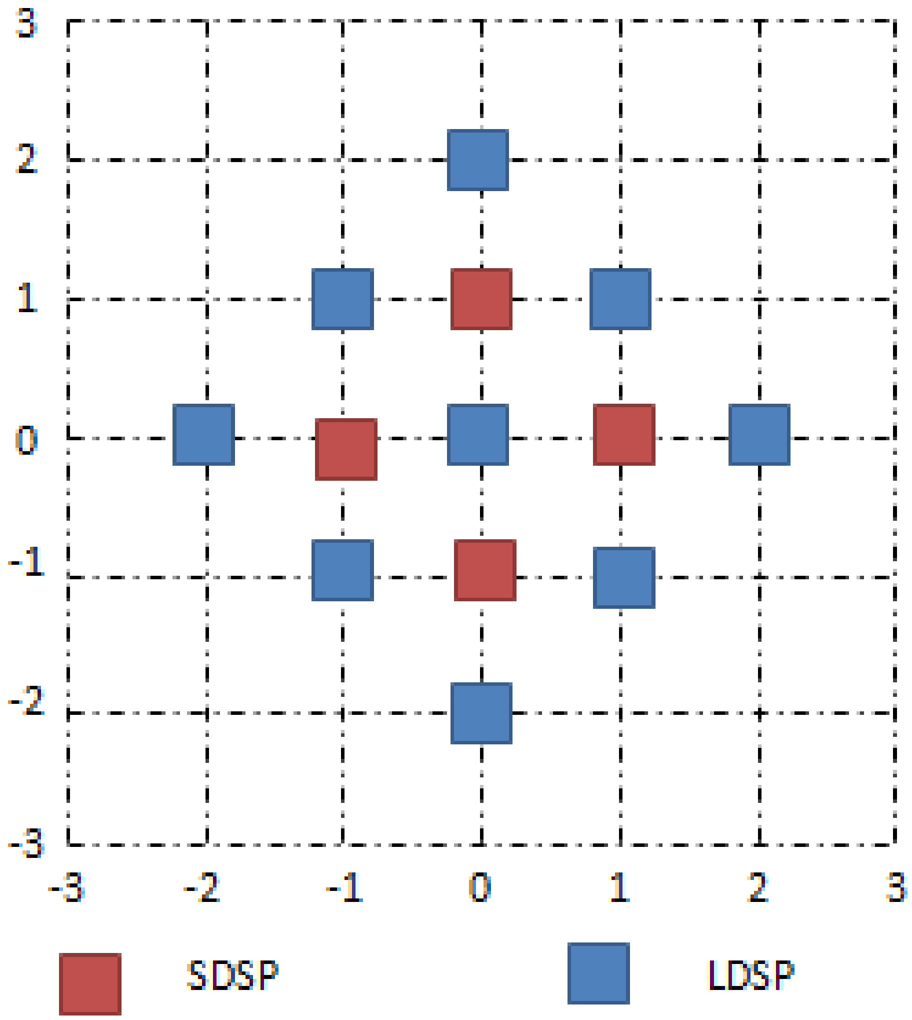


Fig. 3. Large Diamond Search Pattern (LDSP) and Small Diamond Search Pattern (SDSP)

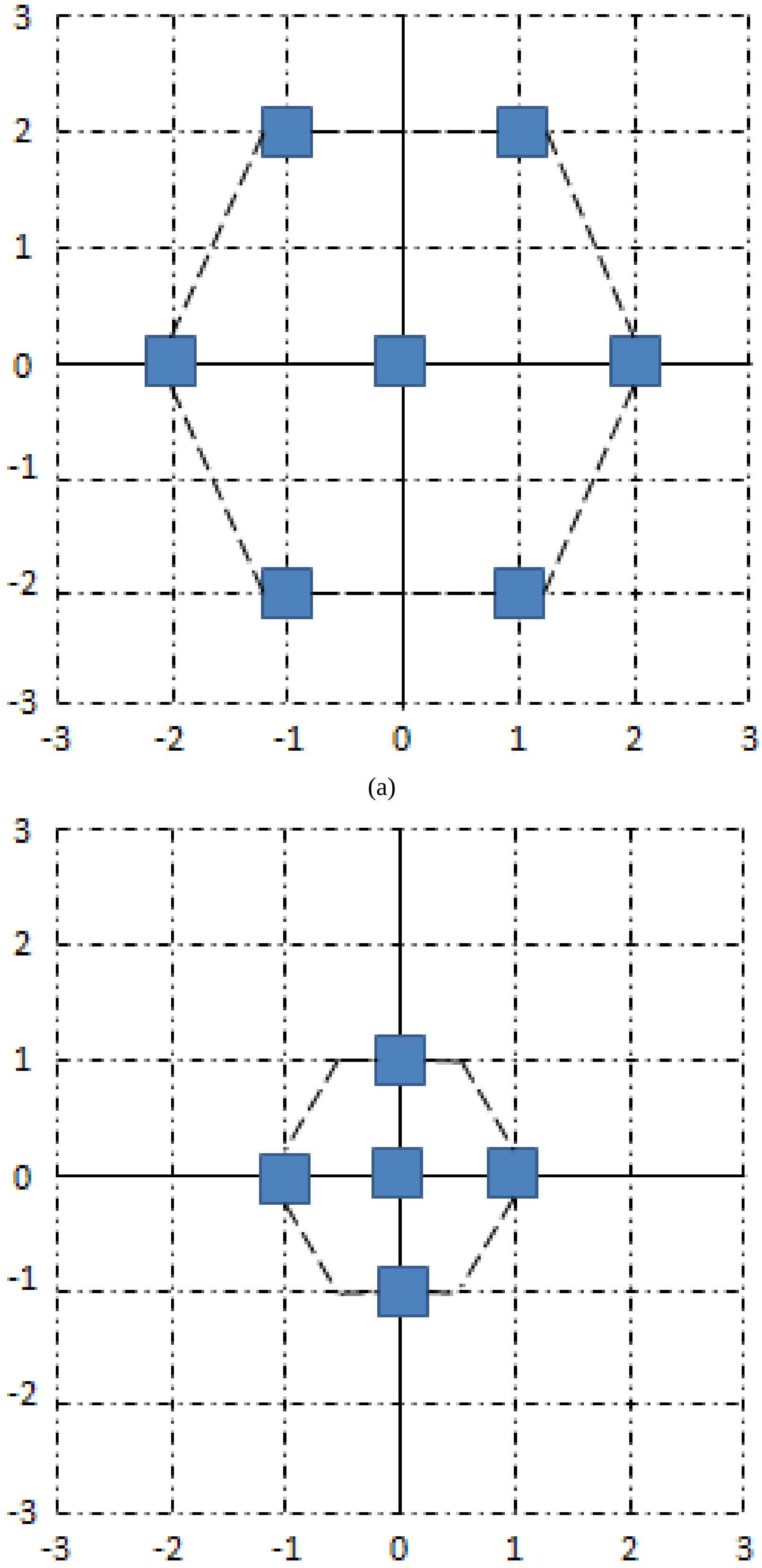
B. Hexagon Search(HS)

Zhu et al. [9] investigated the research and proposed a Hexagon based search algorithm that achieves better speed improvement up than DS algorithm with similar distortion performance [4]. A Large hexagon search pattern (LHSP) consists of seven checking points with the center surrounded by the six hexagon endpoints. For the best match at the location (3,-1), the HS algorithm initially uses the Large Hexagon Search Pattern (LHSP) as shown in Fig. 4(a) with seven checking points is centered at (0, 0) in the motion field. With the minimum MSE point at the location (2, 0) as the center, a new large hexagon is formed with the three new candidate points. With (3,-2) as the center point, a new hexagon is formed. Since this point is still the minimum distortion point, switch the search pattern from the large to the Small Hexagon Search Pattern (SHSP) as shown in Fig. 4b. The four points covered by the small hexagon are evaluated. The new minimum MSE point is the final solution of the motion vector which is obtained at the location (3,-1) with 17 search points as shown in Fig. 5.

C. New Hexagon Search (NHEXS)

The NHEXS algorithm in [6] uses Small Cross Search Pattern (SCSP) and Large Cross Search Pattern (LCSP) as initiative steps in the algorithm. The SCSP and LCSP are illustrated in Fig. 6. This will guide the possible correct direction for the HS pattern in subsequent steps to find the absolute motion vector.

Considering the best match to be at (3, -1), the NHEXS algorithm utilizes 5 search points of SCSP pattern to find the minimum block difference in the initial step. A new SCSP is formed by using the vertex (1, 0) as the center. The three unchecked outermost search points of the LCSP and the two unchecked points of the SCB (radius = ±2) are checked. This will guide the possible correct direction for the LHSP pattern to be formed at (2, 0) and in subsequent step SHSP is used finally with center point at (3, -2) to obtain the best match at the location (3,-1). The algorithm utilizes 21 search points as shown in Fig. 7 to find the exact match.



(a)

(b)

Fig. 4. (a) Large Hexagon Search Pattern (LHSP) (b) Small Hexagon Search Pattern (SHSP)

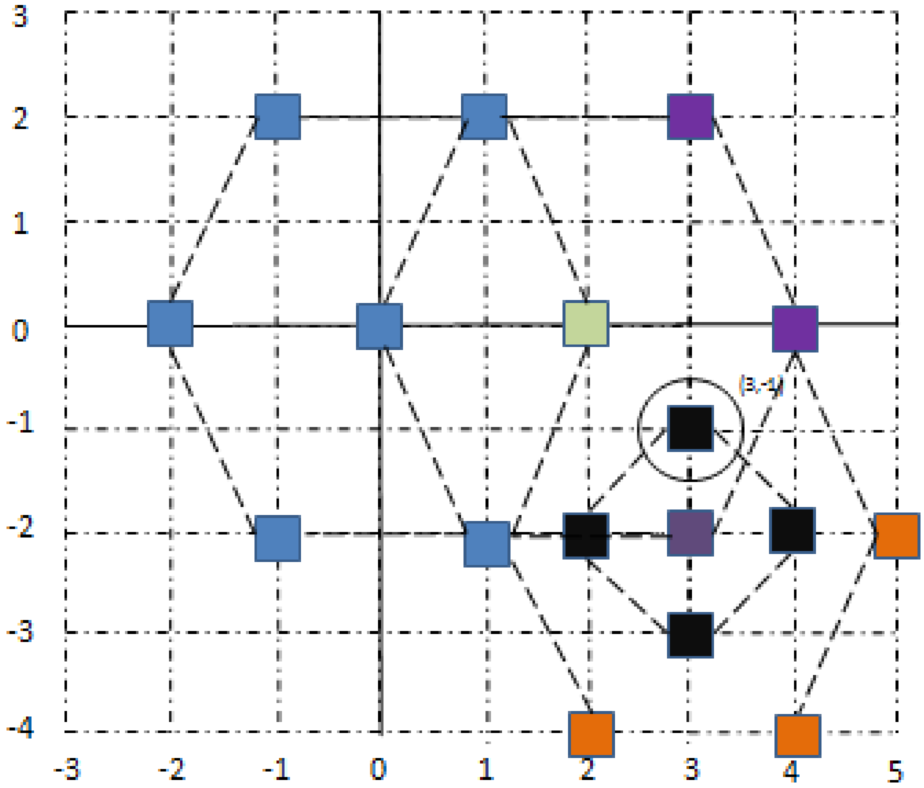


Fig. 5. Search path example for HS with best match at location (3,-1)

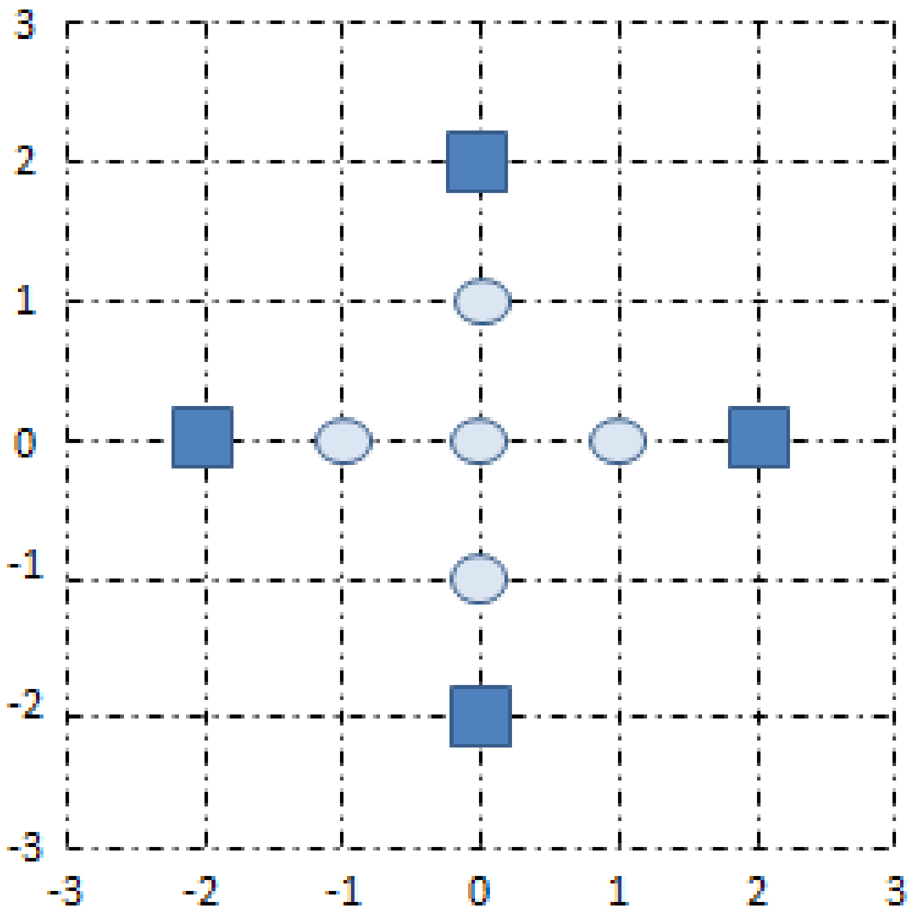


Fig. 6. Small Cross Search Pattern (SCSP) and Large Cross Search Pattern (LCSP)

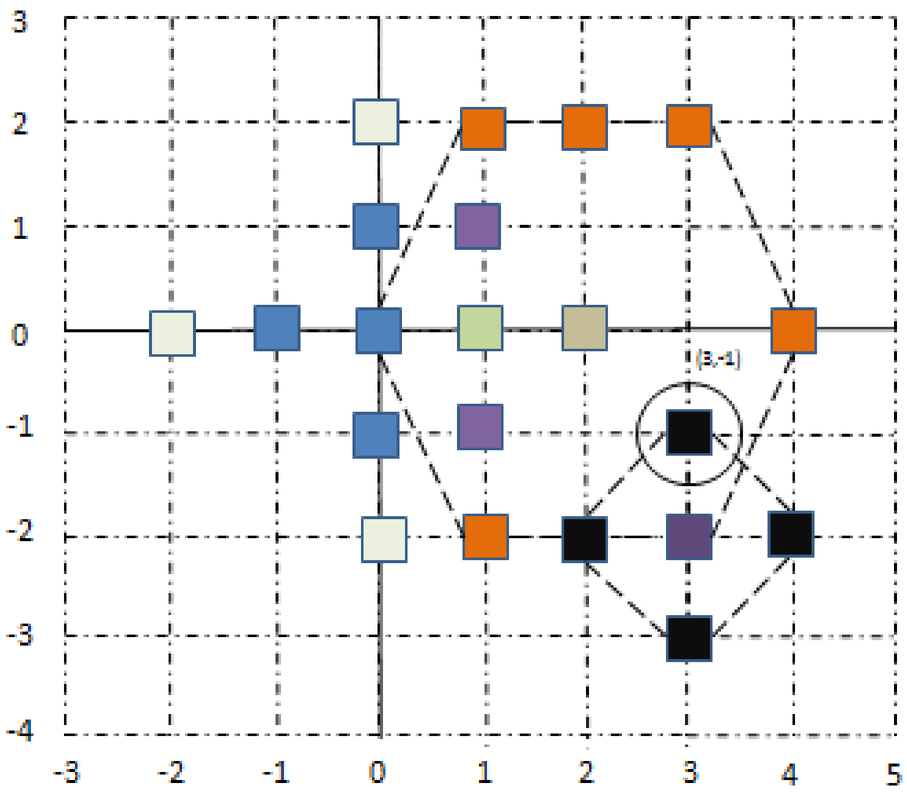


Fig. 7. Search path example for NHEXS with best match at location (3,-1)

IV. Proposed Algorithm

Hexagon pattern helps reducing the computational complexity to a great extent, but there is still complexity in coding. Taking this in account, a new hybrid search pattern is designed so as to reduce the computational complexity. It adopts three different patterns at each step to achieve the minimum number of search points. The algorithm also concentrates on reduction of ME complexity with comparable performance. The following section will explain the way by which the hybrid algorithm is developed using the available patterns.

A. Algorithm Development

The algorithm is designed using Symmetric Cross pattern with step size ±2, modified asymmetric Hexagon with step size ±1 and small Diamond pattern with step size ±1 from center point. The steps for proposed algorithm are given below:

Step 1: The search starts with the Symmetric Cross search pattern with pixel distance ± 2 as shown in Fig.8. The minimum cost point is obtained using Mean Absolute Difference computation. If the minimum cost is found to be at center point, then go to Step 3.

Step 2: The Modified Asymmetric Hexagon search is performed with pixel distance ± 1 as shown in Fig. 9 as second search with minimum cost point from Step 1 as center point.

Step 3: Switch from the coarse search to the fine resolution inner search to find the best motion vector. With minimum point from previous search as center point, the search is done with small Diamond pattern having step size ±1 as shown in Fig.10.The resultant minimum point from this step will be used to calculate the motion vector.

The search starts from initial point (0, 0) as the center point and first search is done using Symmetric Cross pattern with step size ±2 and five search points are initiated. With the minimum cost point obtained, the second search is performed with Modified Asymmetric Hexagon Search with step size of ±1 and using seven search points. The center point is reused from the first search. The minimum cost point from second search is the center point for the final search. The final search is performed using small diamond search with five search point with step size of ±1. Here three points are reused and the minimum cost point obtained is the best match block for the current block and motion vector is calculated with reference to this point. The proposed MCHDS algorithm adopts the following pseudocode to find the minimum cost:

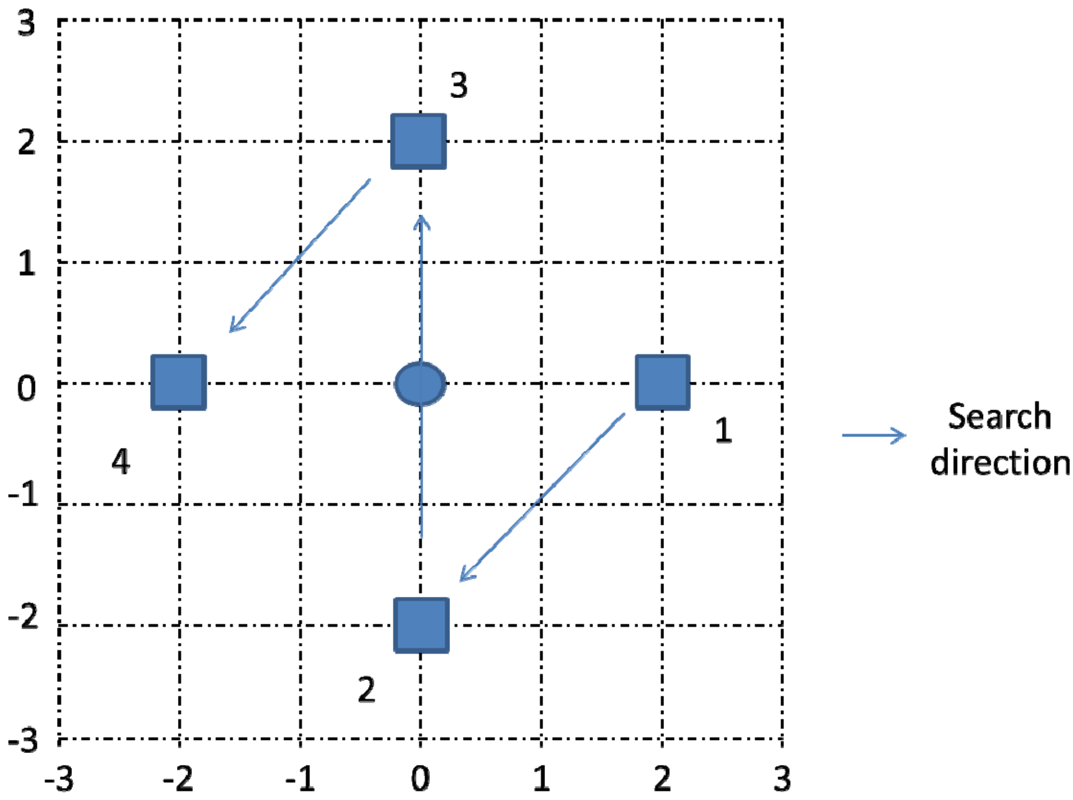


Fig. 8. First Search

Perform initial Cross Search
Centre Point <= Minimum point of Cross Pattern
points<=7
i<= 0
Repeat
MAD(i) <= Execute Modified Hexagon search points
i++
while(i < points)
Minimum cost <= minimum(MAD(i))
Centre Point <= Minimum point of Modified Hexagon Pattern
Perform Small Diamond Search
Minimum cost <= minimum(MAD(i))
Generate vector(Minimum cost)

B. Search Path Analysis

Let us assume that the best match of the current block is at the position (3,-1) as shown in Fig.11. The Symmetric Cross search is first performed and the minimum cost is found using MAD which will be at (2, 0). Having (2, 0) as the center point for modified asymmetric Hexagon search, the search will have 7 search points and the minimum cost point is found to be at (2,-1). With (2,-1) as center point, Small Diamond search is performed and the final minimum cost point will be at (3,-1). Thus the proposed algorithm takes a total of 13 computations to find the best match. For the same assumption, the Hexagon Search (HS) Algorithm proposed by Ce Zhu et al [9] requires 17 computations to find the best match and NHEXS algorithm proposed by Kamel Belloulata et al [6] requires 21 computations to find its best match. Thereby, the number of search points is reduced which in turn reduces the motion estimation time of the proposed algorithm to a great extent.

To evaluate the performance of the new CHDS algorithm featured with inner search, we compare it against the existing methods in terms of number of search points, average computational time savings for motion estimation and Peak Signal to Noise Ratio (PSNR). The experimental setup is as follows: the block size of 16 x16, the maximum displacement in the search area is ± 2 pixels and MAD (Mean absolute Difference) for distortion measurement. Encoding is done for different test sequences with different resolutions. As motion estimation consumes more time in total encoding process, the time taken for motion estimation to predict the motion vector is analysed. The existing and the proposed algorithms are simulated using MATLAB 7.10.0.499 (R2010a). The maximum CU size of 64 with maximum partition of depth 2 is taken. The algorithm is applied to the first 30 frames of sequences “Foreman” with a resolution of 352x288, “Mobile” with a resolution of 352 x 288, “BQMALL” with a resolution of 832 x 480, “MOBISODE” with a resolution of 832x480, “People on street” with a resolution of 2560x1600 and “Traffic” with a resolution of 2560 x 1600 . These sequences vary in motion content as well as in frame size.

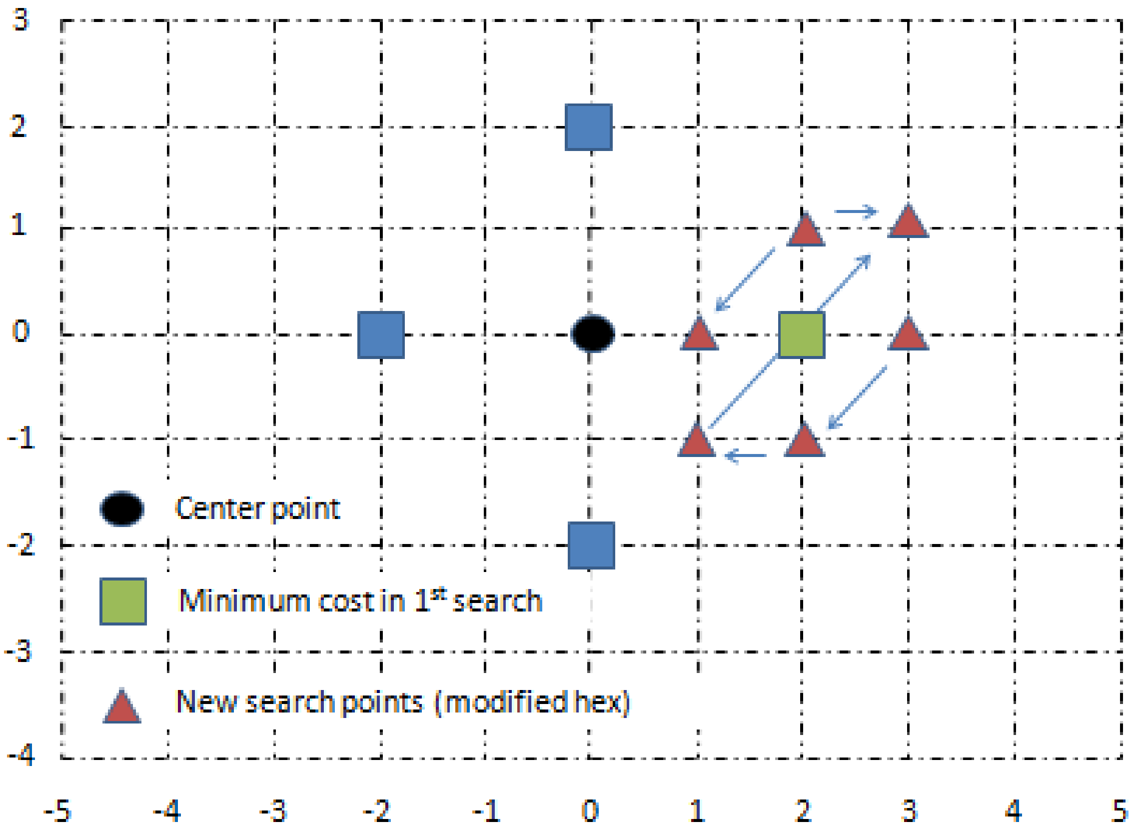


Fig. 9. Second Search

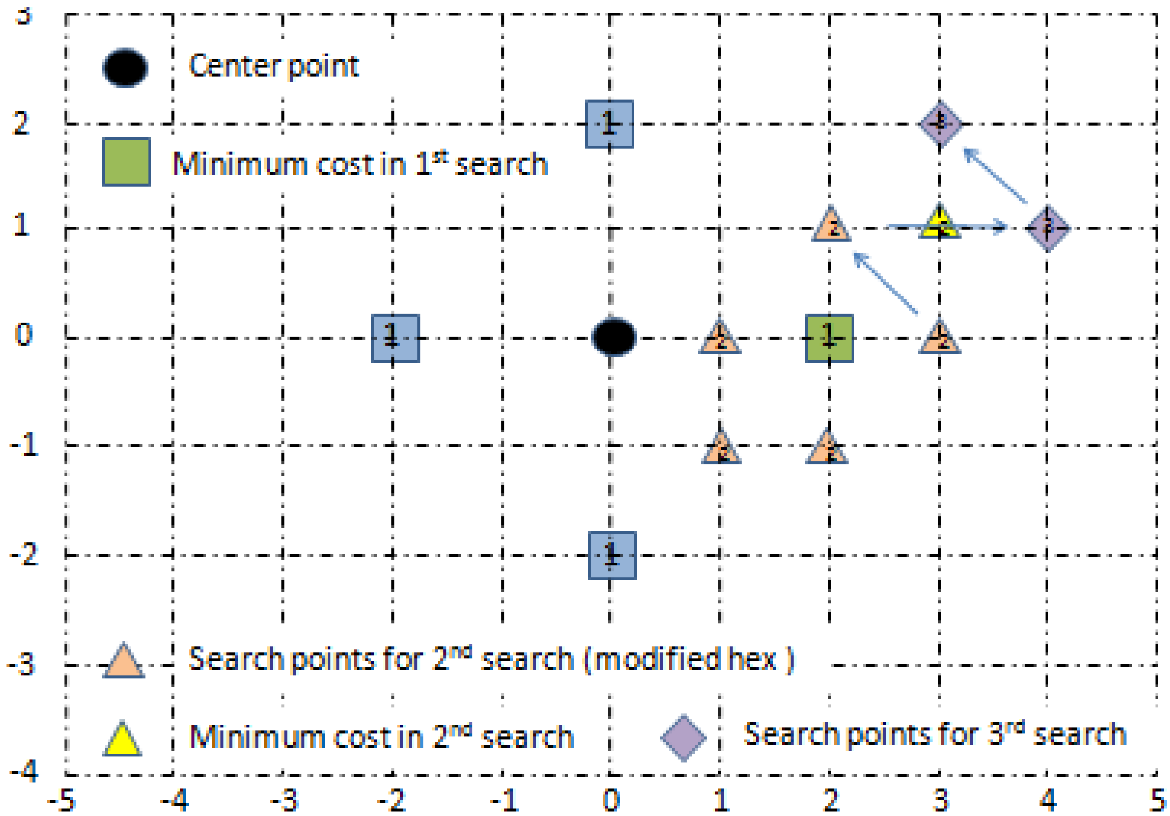


Fig. 10. Third Search

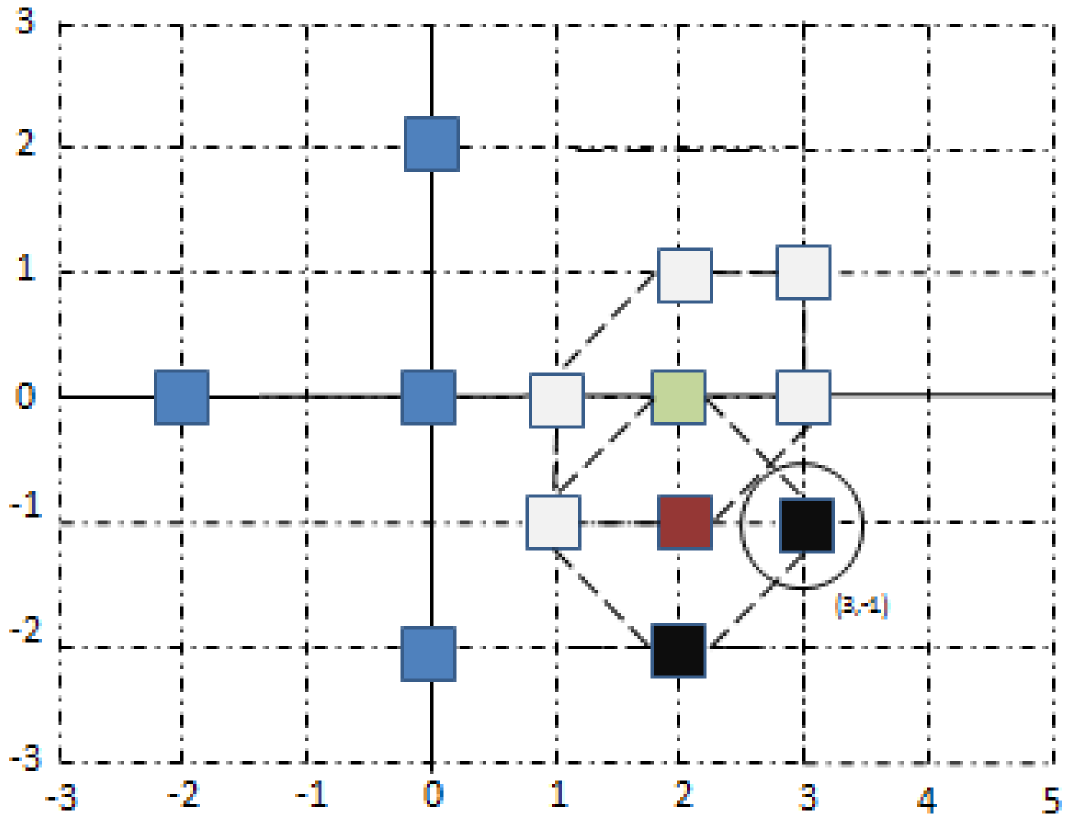


Fig. 11. Search Path Analysis of Proposed Algorithm

V. EXPERIMENTAL RESULTS

Simulations are carried out on Windows 8.1 OS platform with Intel I3-2310M @ 2.10 GHz CPU and 4GB RAM. Motion Estimation time is calculated for different sequence for encoding 30 frames. Test sequences are put into test with maximum block size of 16x16. Table I shows the simulation results of Average ME time for different search patterns such as Diamond search (DS), Hexagon Search (HS), New Cross Hexagon search (NHEXS) and proposed search algorithm. It can be clearly seen that there is an appreciable decrease of average ME Time compared to other existing search algorithms.

TABLE I
SIMULATION RESULTS FOR AVERAGE ME TIME

Sequence	ME Time (sec) for 30 frames			
	DS	**HS**	**NHEXS**	**Proposed**
Foreman	19.38798	15.23368	19.86043	12.34941
Mobile	16.16917	13.32104	19.88173	11.29907
Bqmall	61.20704	48.77680	54.76506	41.98951
Mobisode	77.94544	57.70776	69.42224	44.32462
People On Street	61.20704	48.77680	54.76506	41.98951
Traffic	77.94544	57.70776	69.42224	44.32462
Average ME Time	52.31035	40.25397	48.01946	32.71279

The ΔME Time and ΔPSNR denotes the CPU time reduction in ME process and improvement of Peak Signal to Noise Ratio (PSNR) of the encoded image respectively. They are defined as in equations (2), (3) and (4):

$$\Delta METime = MET_{proposed} - MET_{original} \quad (2)$$

$$\%\Delta METime = \frac{\left(MET_{original} - MET_{proposed}\right)}{MET_{original}} \times 100 \quad (3)$$

$$\Delta PSNR = PSNR_{proposed} - PSNR_{original} \quad (4)$$

where MET proposed and PSNRproposed denotes the ME time and PSNR of the proposed algorithm. The METoriginal and PSNRoriginal represents the ME time and PSNR of DS, NHEXS and HS.

Table II shows the improvement in PSNR and average reduction of ME Time for proposed algorithm when compared to existing DS, HS and NHEXS search patterns. It is clear that the proposed MCHDS algorithm reduces an average of 18.05% of Motion Estimation time compared to Hexagon Search (HS), 35.91% of time for motion estimation compared to Diamond Search (DS) and 33.32% of Motion Estimation time compared to New Hexagon Search (NHEXS).

The comparison of the existing and the proposed algorithms is shown in Fig.12 for thirty frames of different video sequences. When compared to the existing algorithms, the proposed scheme can reduce the ME time of the encoded image, as the frame number increases.

Average search points for each block in a frame are calculated for different sequence in encoding 30 frames. Table III shows that the proposed CHDS algorithm consumes the smallest number of search points compared to other algorithms. The average search points per block with the observations are MCHDS<<HS<<NHEXS<<DS.

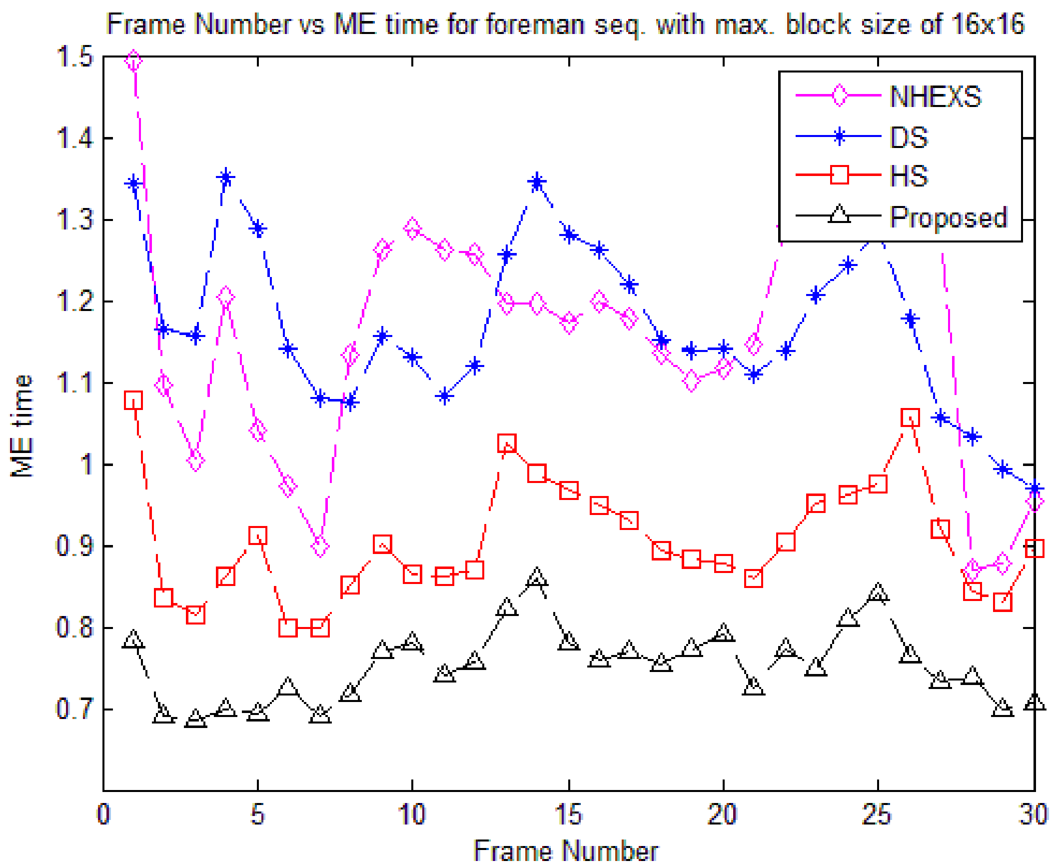
In order to demonstrate the performance of the proposed algorithm, the Fig.13 gives the result of the average number of computations per block for different video sequences as the frame number increases. It clearly manifests the superiority of the proposed CHDS algorithm to the other methods in terms of number of search points used.

TABLE II
SIMULATION RESULTS SHOWING PSNR IMPROVEMENT AND AVERAGE REDUCTION OF ME TIME

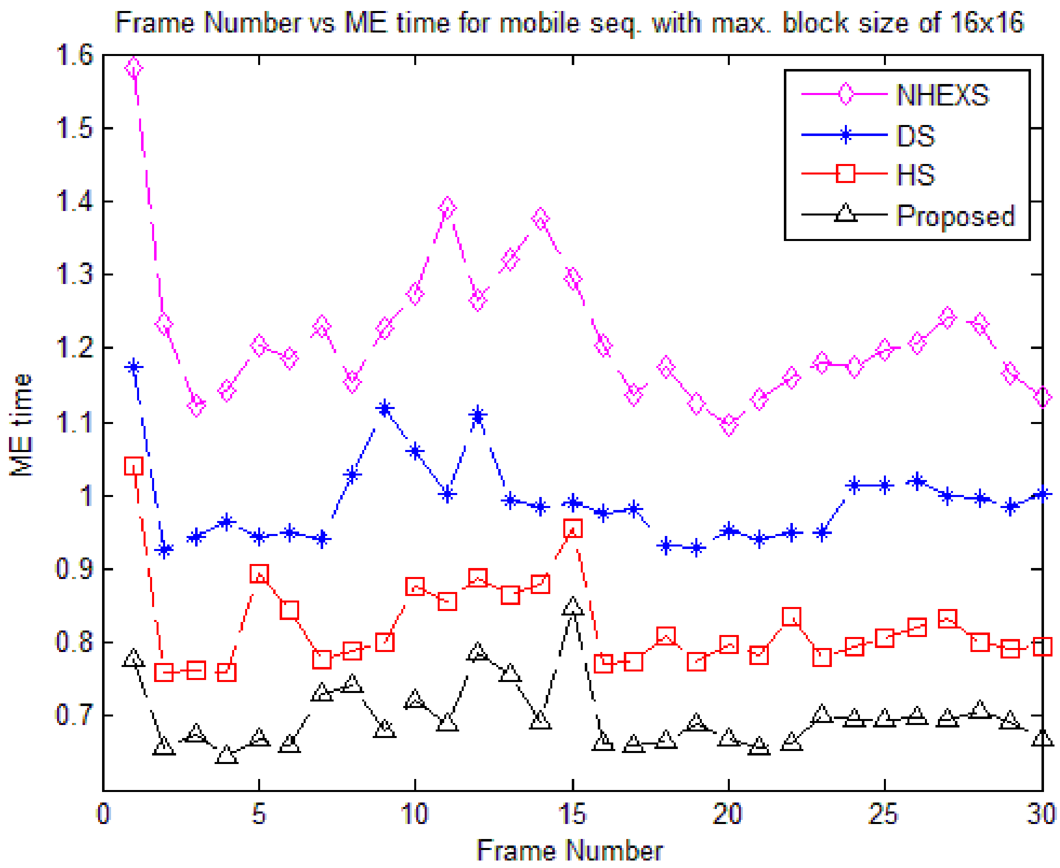
Sequence	**PSNR Improvement (dB) (ΔPSNR)**			**ΔME Time (sec)**		
	Proposed VS DS	**Proposed VS HS**	**Proposed vs NHEXS**	**Proposed vs DS**	**Proposed vs HS**	**Proposed vs NHEXS**
Foreman	2.50393	0.04237	3.94352	-7.03858	-2.88427	-7.51102
Mobile	3.09627	-0.38212	4.92830	-4.87010	-2.02196	-8.58266
Bqmall	0.41188	0.11444	0.79441	-19.21753	-6.78730	-12.77555
Mobisode	1.66962	-1.02872	0.90358	-33.62082	-13.38314	-25.09763
People On Street	0.77601	0.24386	0.59925	-250.36850	-99.46161	-203.35108
Traffic	0.16458	0.03065	0.50600	-171.33863	-72.20404	-98.81985
Average % time savings of the Proposed Algorithm				35.91436	18.05412	33.32462

TABLE III
AVERAGE NUMBER OF SEARCH POINTS PER BLOCK

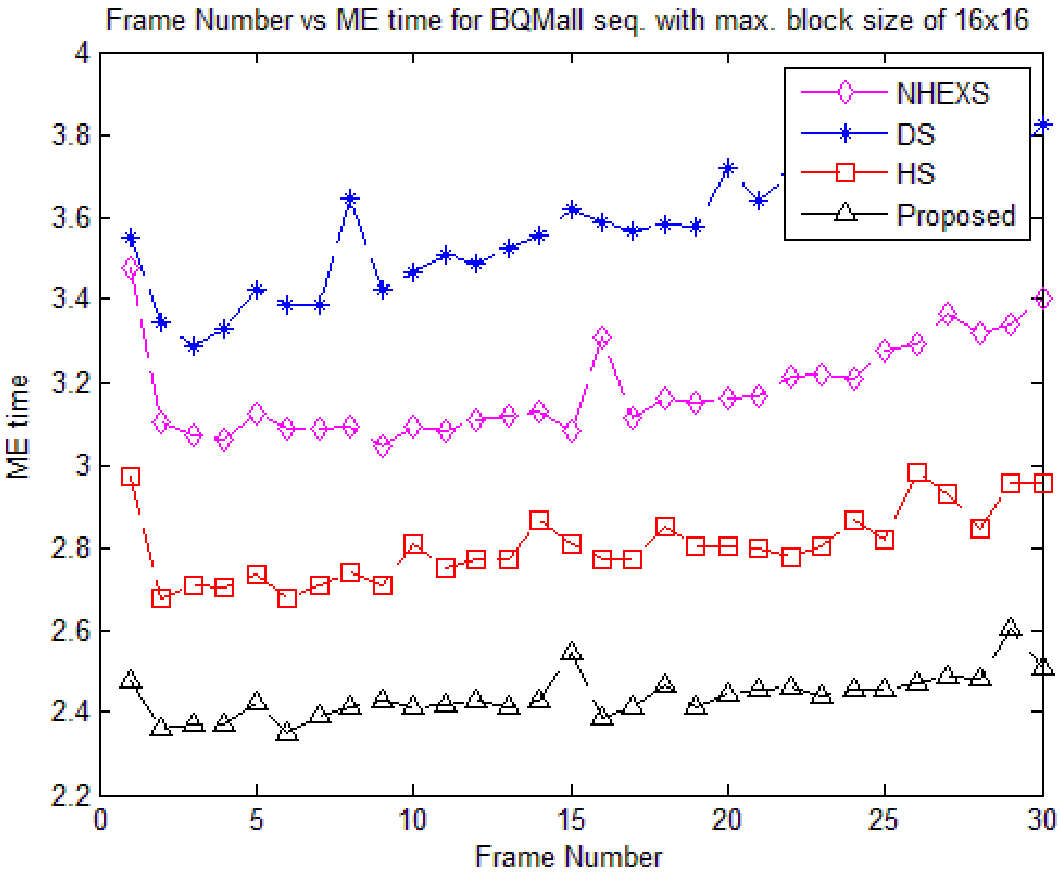
Sequence	**Average no. of Search points (computations) for each block**			
	DS	HS	NHEXS	Proposed
Foreman	16.83942	12.08405	12.71671	9.48088
Mobile	14.32350	11.04308	12.73074	8.99678
Bqmall	15.51905	11.43994	11.20496	9.54289
Mobisode	18.79306	12.98219	13.31057	9.87241
People On Street	17.57339	12.20654	12.67362	9.52135
Traffic	14.90163	11.55389	10.69535	9.35741
Average Computations	16.32501	11.88495	12.22199	9.461953



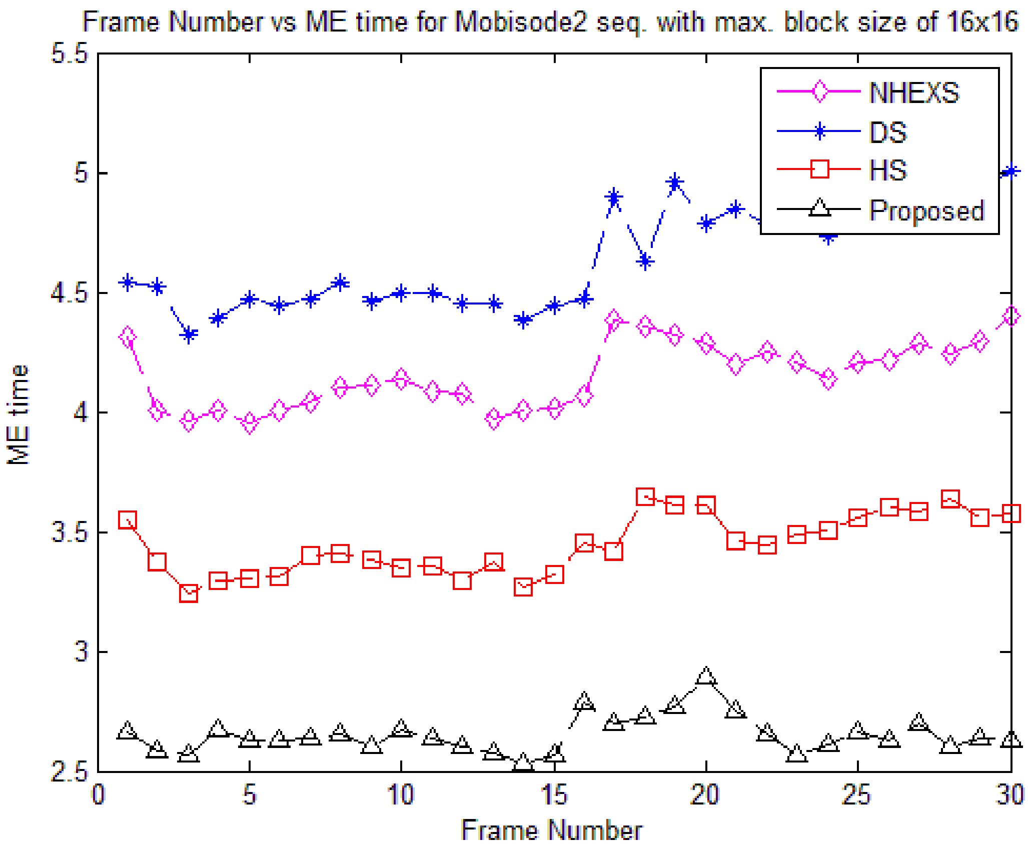
a. Foreman sequence



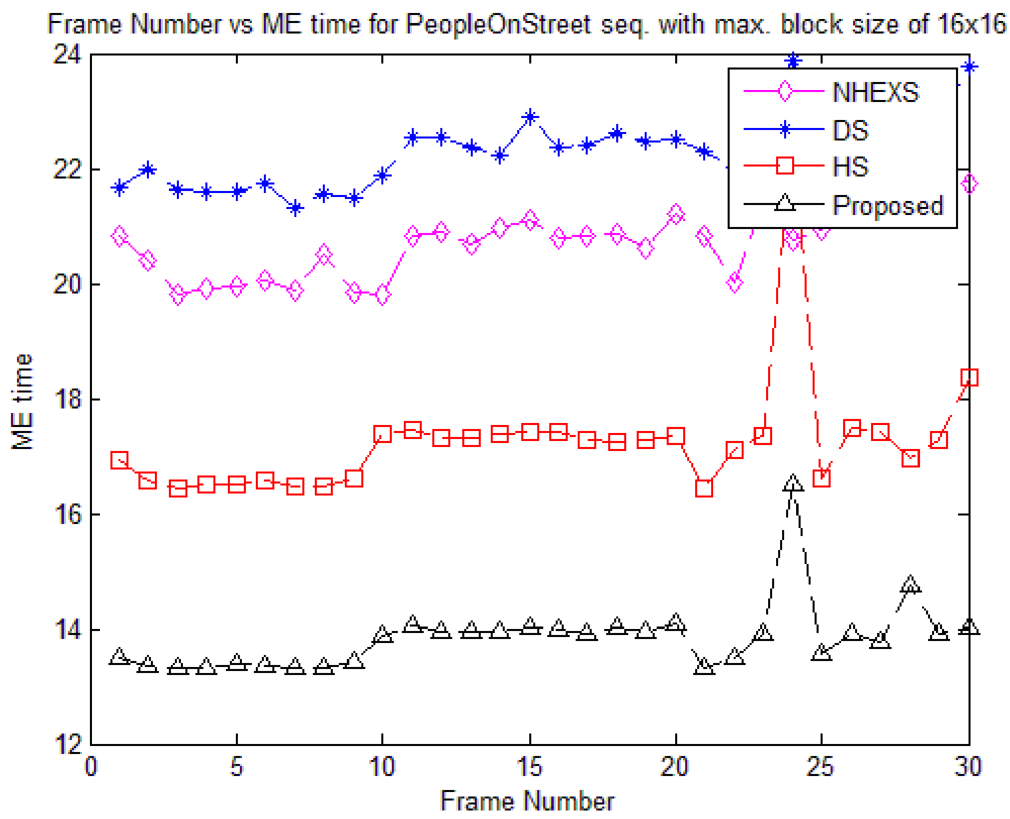
b. Mobile sequence



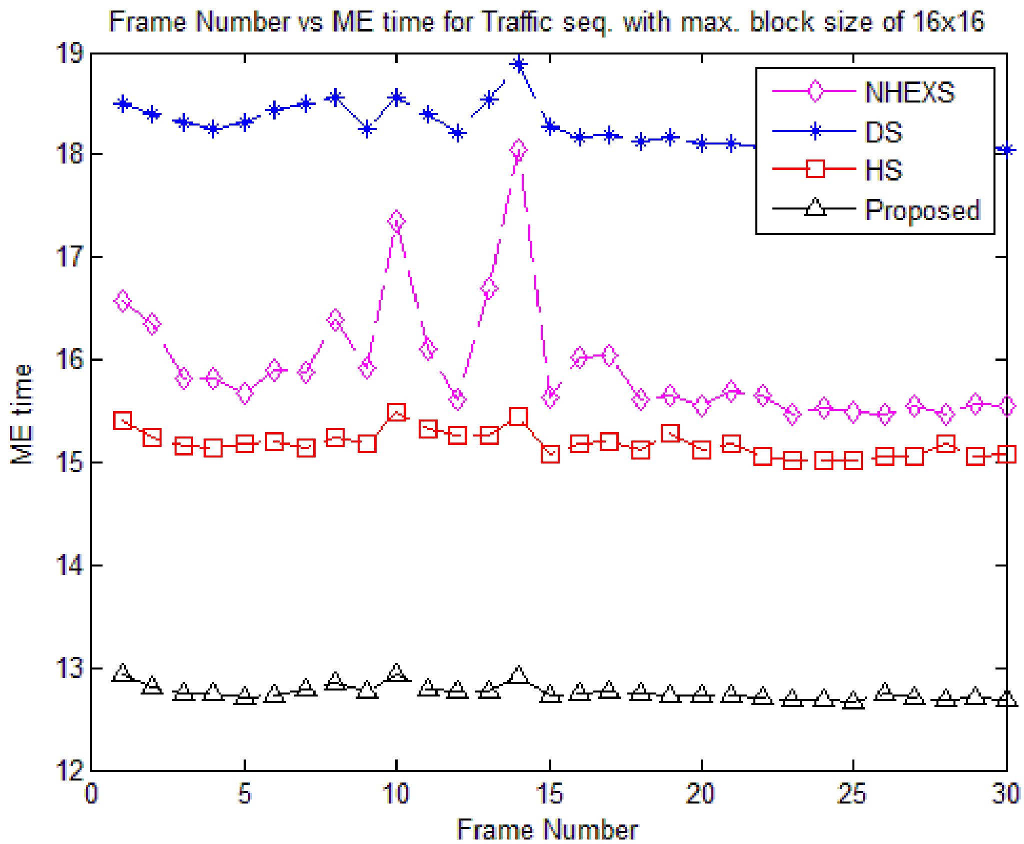
c. BQMall sequence



d. Mobisode sequence

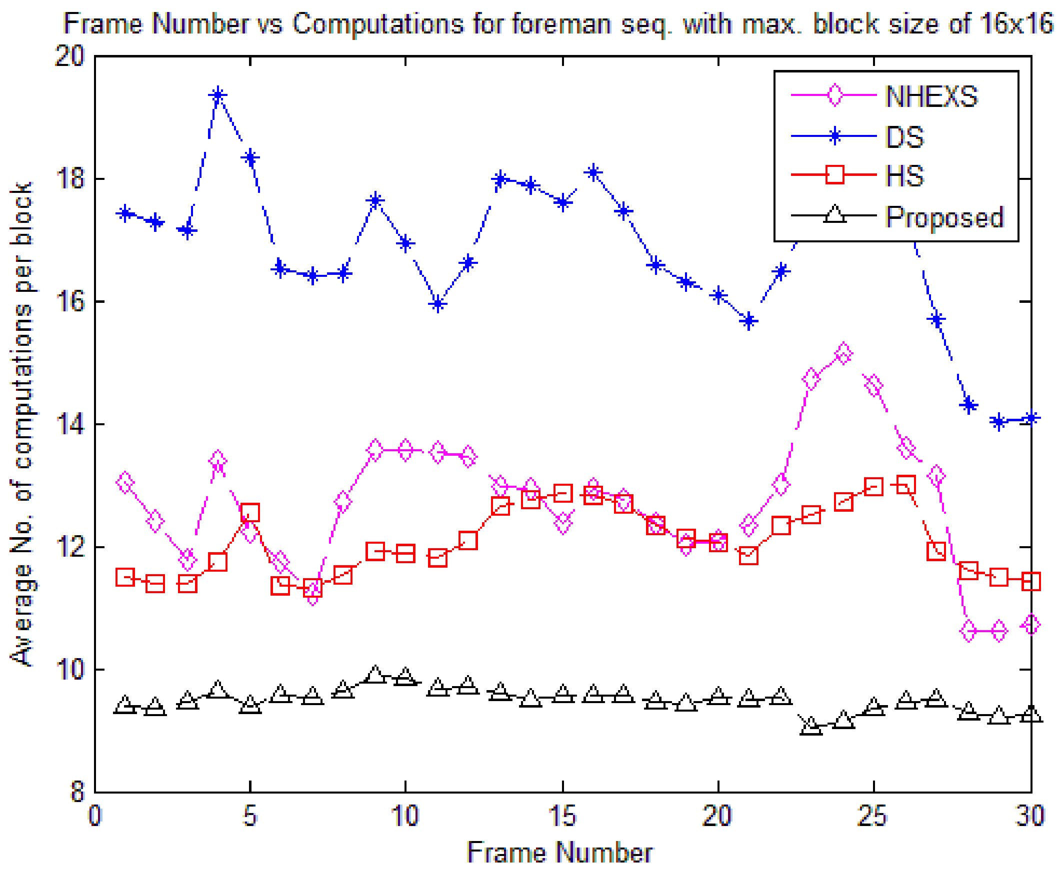


e. People on street

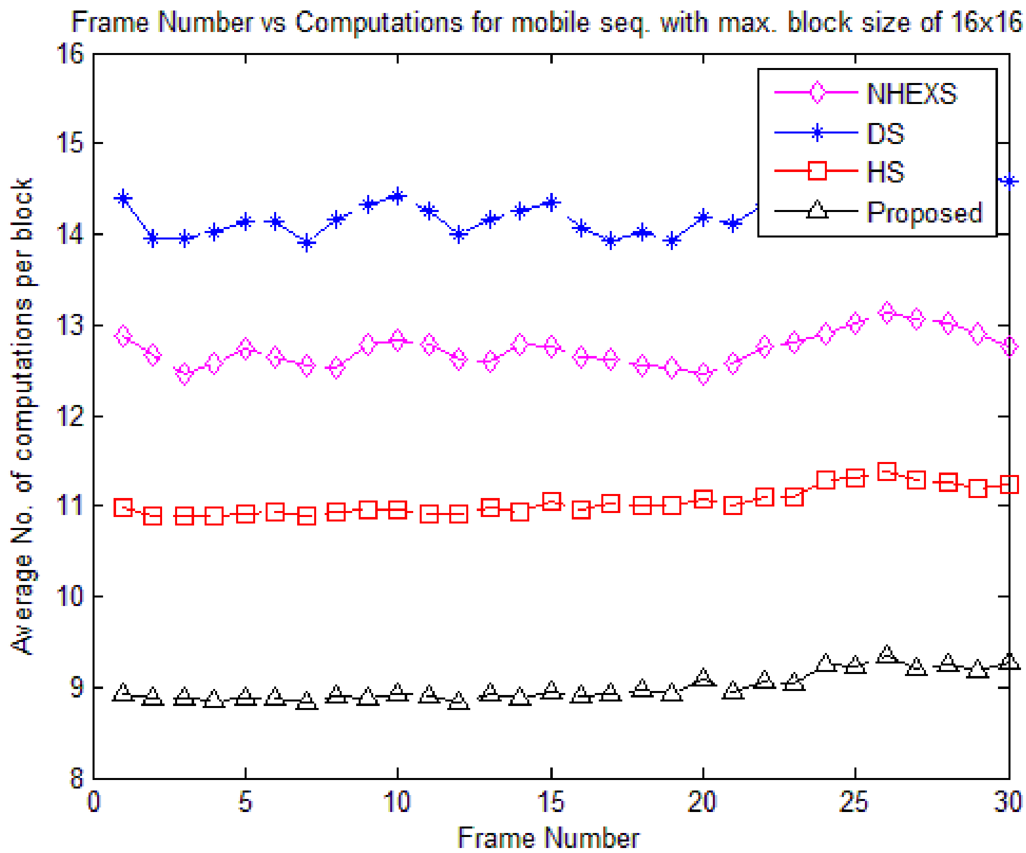


f. Traffic

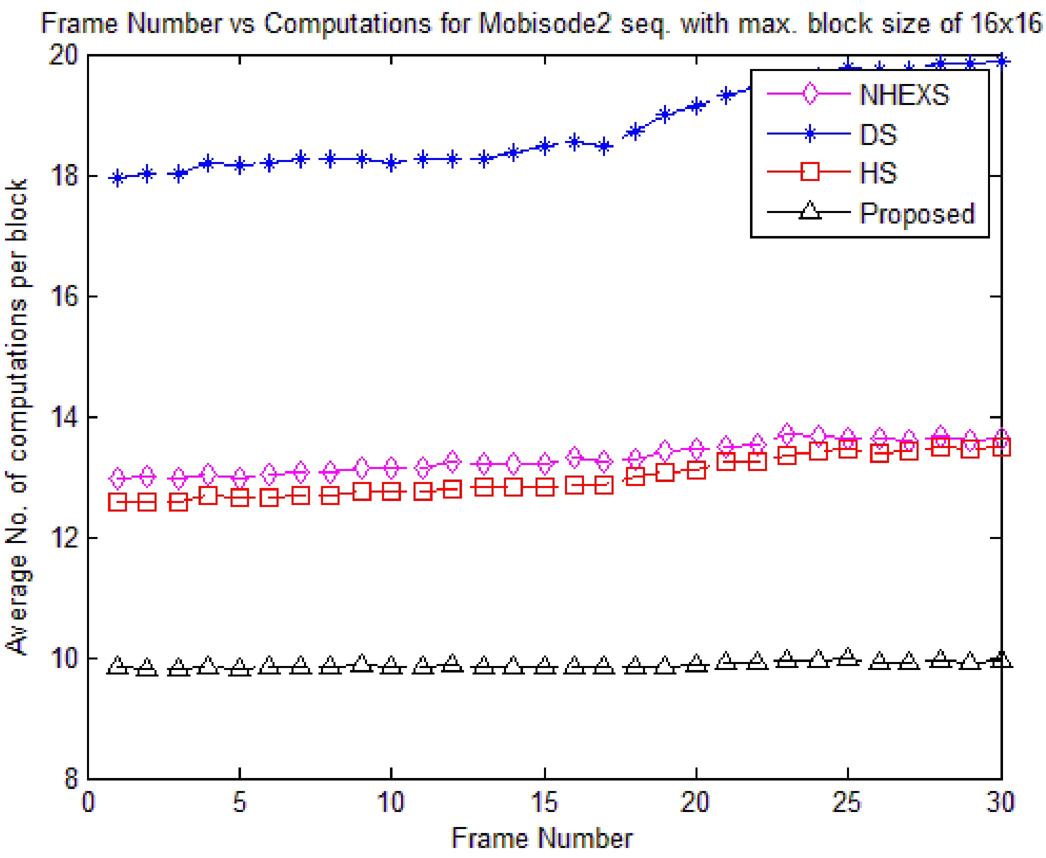
Fig. 12. Performance comparison of DS, HS, NHEXS and the proposed CHDS for (a). Foreman Sequence (b). Mobile sequence (c). Mobisode sequence (d). BQMALL sequence (e). People on street sequence (f). Traffic sequence in terms of time reduction in ME process.



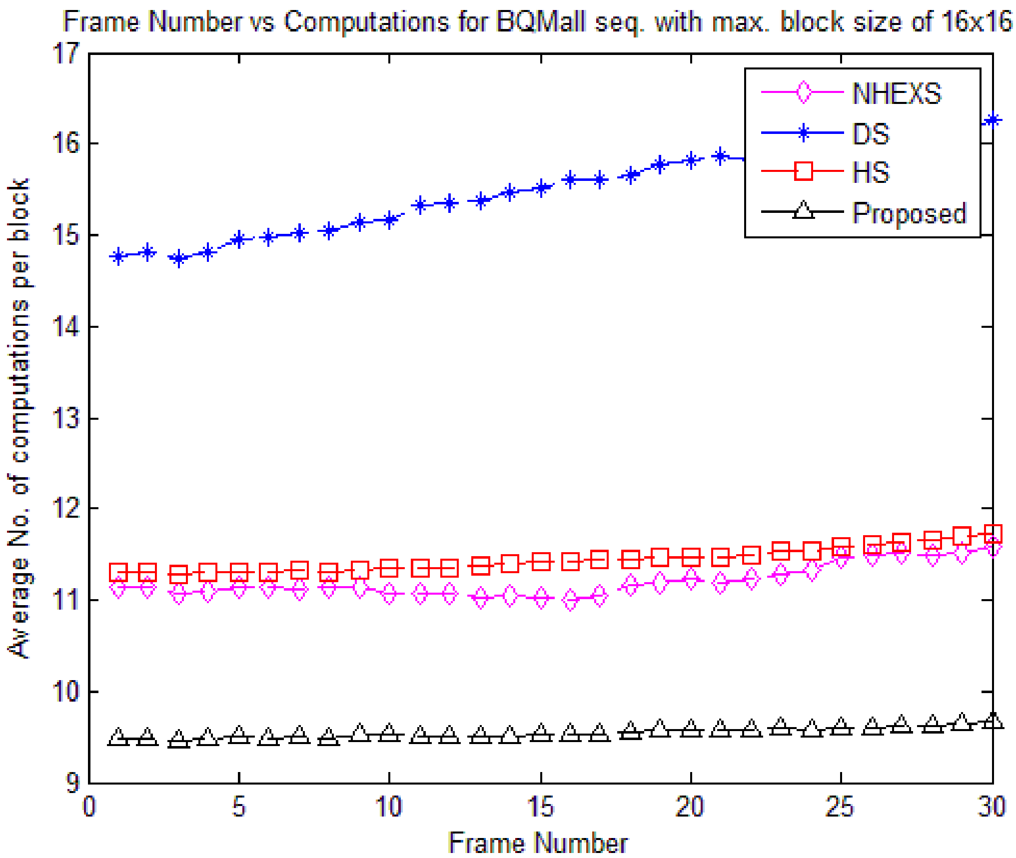
a. Foreman sequence



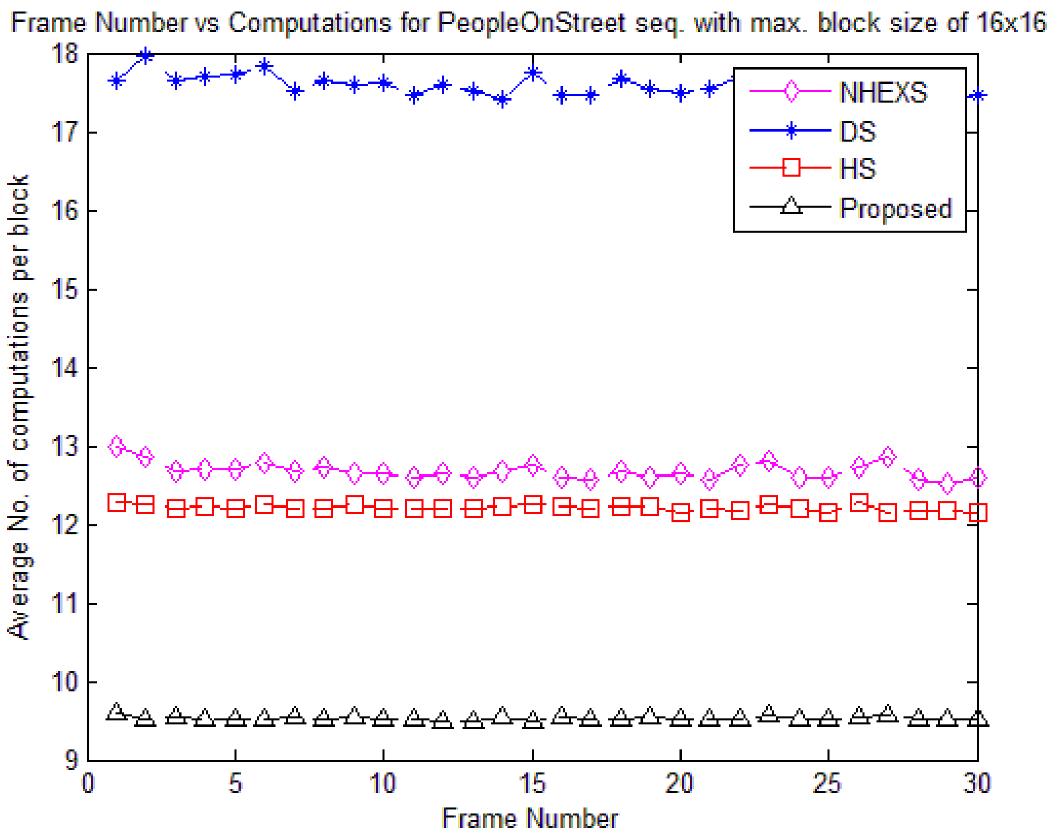
b. mobile sequence



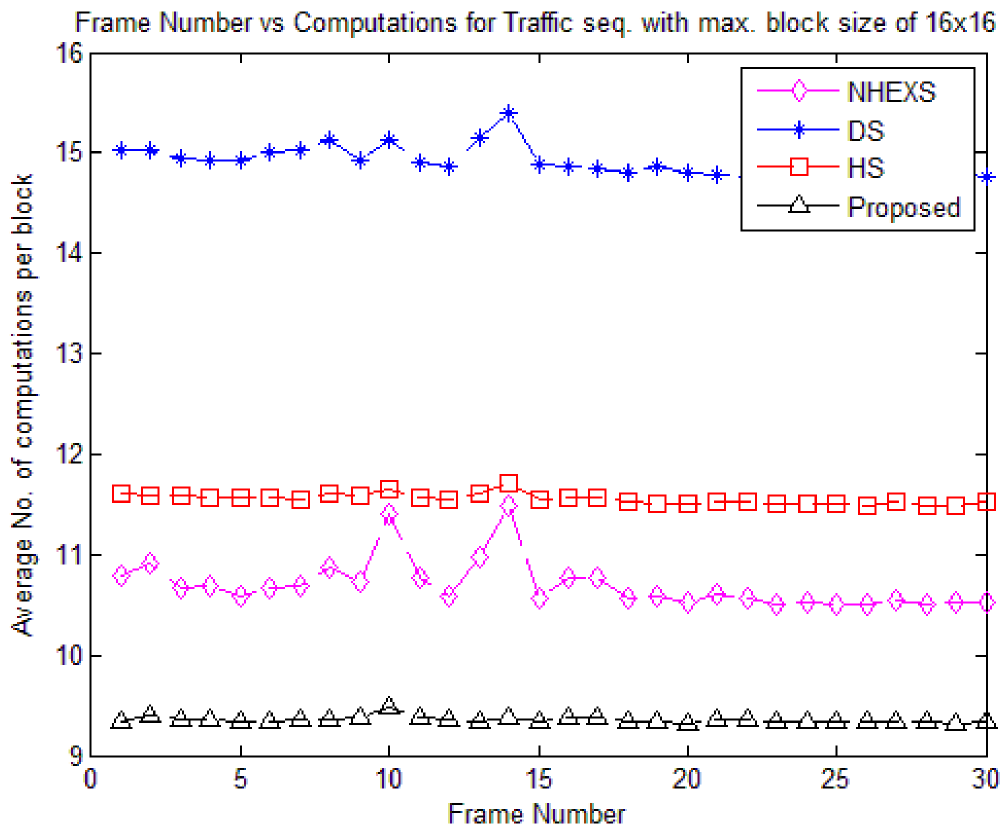
c. mobisode sequence



d. BQMall sequence



e. People on street



f. Traffic

Fig. 13. Performance comparison of DS, HS, NHEXS and the proposed CHDS for (a) Foreman Sequence (b) Mobile sequence (c) Mobisode sequence (d) BQMALL sequence (e) People on street sequence (f). Traffic sequence in terms of average number of computations per block

VI. CONCLUSION

In this paper, we proposed a new Modified Cross Hexagon Diamond Search (MCHDS) algorithm for fast Motion Estimation and analysed the performance related to ME Time, average number of search points and PSNR. The simulation results reveal that the proposed algorithm reduces an average of 33.32% of time for motion estimation compared to NHEXS, 18.05% of time for motion estimation compared to HS and 35.91% of time for motion estimation compared to DS, for motion in video sequences with a block size of 16x16. The proposed algorithm also shows an appreciable improvement in quality and reduction in number of computations per block when compared to the previous algorithms. The experimental results have demonstrated that the proposed MCHDS algorithm outperforms the previous algorithms and it works well in different formats of video sequences with various motion activities.

REFERENCES

[1] Ce Zhu, Xiao Lin, Lappui Chau, and Lai-Man Po , "Enhanced hexagonal search for fast block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 14, no. 10, pp. 1210-1214, 2004.

[2] Obianuju Ndili and Tokunbo Ogunfunmi, "Algorithm and architecture co-design of hardware-oriented, modified diamond search for fast motion estimation in H.264/AVC," IEEE Transactions on Circuits and Systems for Video Technology, vol. 21, No. 9, pp. 1214-1227, 2011.

[3] Kamel Belloulataa, Shiping Zhub, Jun Tianb and Xiaodong Shen, "A novel cross-hexagon search algorithm for fast block motion estimation," 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA), 2011, pp-1-4.

[4] Akash Ambulkar, Shailesh Wankhede, Sujit Kautkar, Mukesh Angrakh, and Vidya N. More, "Enhanced directional hexagonal search algorithm," IEEE International Conference on Circuits and Systems (ICCAS), October 3-4, 2012, pp.143 – 146.

[5] Ziyin Li and Qi Yang, "A fast adaptive motion estimation algorithm," International Conference on Computer Science and Electronics Engineering, 2012, pp. 656-660.

[6] Kamel Belloulata, Shiping Zhu, and ZaikuoWang , " A fast fractal video coding algorithm using cross-hexagon search for block motion estimation," International Scholarly Research Network Signal Processing , 2011, pp 1-10.

[7] G. J. Sullivan, J.-R. Ohm, W.-J. Han, 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.

[8] 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, No. 1, pp 39-45, 2012.

[9] Ce Zhu, Xiao Lin, Lap-Pui Chau, Keng-Pang Lim, Hock-Ann Ang and Choo-Yin Ong , "A novel hexagon-based search algorithm for fast block motion estimation," Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 03 , 2001, pp. 1593-1596.

[10] Zhaoqing Pan Sam Kwong , "A direction-based unsymmetrical-cross multi-hexagon-grid search algorithm for H.264/AVC motion estimation," Journal of Signal Processing System , pp 59-72 , 2013.

[11] S. Rukmani Devi, P. Rangarajan, and J. Raja Paul Perinbam , "VLSI implementation of high performance optimized architecture for video coding standards," Acta Polytechnica Hungarica, vol. 10, no. 6, pp 237- 249, 2013.

[12] Gustavo Sanchez, Felipe Sampaio, Marcelo Porto, Sergio Bampi, and Luciano Agostini, "DMPDS: A fast motion estimation algorithm targeting high resolution videos and its FPGA," International Journal of Reconfigurable Computing, pp- 1-12, 2012.

[13] Chih-Ming Kung, Wan-Shu Cheng, and Jyh-Horng Jeng , "Application of genetic algorithm to hexagon-based motion estimation," Scientific World Journal, pp. 1-12, 2014.

[14] Jaeho Lee, Seongwan Kim, Kyungmin Lim, Jae Hyun Kim and Sangyoun Lee , "Fast intermode decision algorithm based on general and local residual complexity in H.264/ AVC," EURASIP Journal on Image and Video Processing, pp. 1-11, 2013.

[15] Ankita P. Chauhan , Rohit R. Parmar and Shahida G. Chauhan, "Comparative study on diamond search algorithm for motion estimation," International Journal of Engineering Research & Technology ,vol. 1 no. 10, pp- 1-6, 2012.

[16] Li Ren-xiang, Zeng Bing, and Liou M. L, "A new three-step search algorithm for block motion estimation," IEEE Transactions on Circuits and Systems for Video Technology, vol. 4, pp. 438-442, 1994.

[17] T.Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," Proceedings of National Telecommunication Conference, 1981, pp. G5.3.1–G5.3.5.

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

[19] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," IEEE Transactions of Image Processing, vol. 9, no.2,pp. 287–290, 2000.

[20] T.Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Transactions of Circuits and System for Video Technology, vol. 13, no. 7, pp.560-576, 2003.

AUTHOR PROFILE

R. Vani received her B.E. degree in Electronics and Communication Engineering from Bharathiar University, Coimbatore and M.E. degree in Communication Systems from Anna University, Chennai. She is currently pursuing Ph.D. in Anna University, Chennai and also working as Assistant Professor in the Electronics and Communication Engineering Department, Meenakshi College of Engineering, Chennai. She is member of the ISTE and IEICE. Her research interest includes Motion Estimation and Video Coding.

P. Davis received his B.E degree in Electronics and Communication Engineering from Ponjesly College of Engineering, Tamil Nadu, India, in 2012, and the M.E degree in Embedded System Technologies from Karpaga Vinayaga College of Engineering and Technology, Tamil Nadu, India, in 2014. His interest includes Video Coding, Motion estimation algorithms.

M. Sangeetha born in 1975 in Tamilnadu, India, received her B.E. degree from the Bharathidasan University, Trichy, in 1996 and M.E. degree from the University of Madras in 1999. She is currently with Karpaga Vinayaga College of Engineering and Technology in the Department of Electronics and Communication Engineering and Ph.D. degree with the Anna University, Chennai, India. She is a member of the IEEE and ISTE. Her research interests are Hardware/Software Co-design and High Level Synthesis.