INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 12(5), 715-727, October 2017.

# Fast Motion Estimation Algorithm using Hybrid Search Patterns for Video Streaming Application

R. Vani, N. Ushabhanu, M. Sangeetha

**Vani Rajamanickam***
Meenakshi College of Engineering, Chennai, India
*Corresponding author: ecehod@mce.edu.in

**Ushabhanu Nageswaran**
Valliammai Engineering College, SRM Nagar, Chennai, India
ushabhanun.ece@valliammai.co.in

**Sangeetha Marikkannan**
Bharath University, Chennai, India

**Abstract:** The objective of the paper is to develop new block matching Motion Estimation (ME) algorithm using hybrid search patterns along the direction of best match. The search efficiency for sequences with fast motions and high resolutions is improved by proposing New Cross Diagonal-Hexagon Search (NCDHS) algorithm which involves a novel multi half-hexagon grid global search pattern and a cross diagonal-hexagon local search pattern. The new search pattern enables the proposed algorithm to perform better search using 9.068 search points on an average, to obtain optimal motion vector with slight improvement in quality. This inturn reduces ME Time upto 50.11%, 47.12%, 32.99% and 43.28% on average when compared to the existing Diamond Search (DS), Hexagon Search (HS), New Cross Hexagon Search (NHEXS) and Enhanced Diamond Search (EDS) algorithms respectively. The novelty of the algorithm is further achieved by applying the algorithm proposed for live streaming application. The NCDHS algorithm is run on two MATLAB sessions on the same computer by establishing the connection using Transmission Control Protocol (TCP) /Internet Protocol (IP) network. The ME Time obtained is 14.5986 seconds for a block size $16 \times 16$, is less when compared to existing algorithms and that makes the NCDHS algorithm suitable for real time streaming application.
**Keywords:** hybrid search patterns, motion estimation time, search points, peak signal-to-noise ratio (PSNR).

## 1 Introduction

The H.264/Advanced Video Coding (AVC) standard was developed by Joint Video Team (JVT) in the year 2003 [6]. The H.264 standard adopts the Variable Block Size Motion Estimation (VBSME) feature which enables the standard to find its importance in various applications such as High Definition Digital Video Disc (HD-DVD), digital video broadcasting and mobile applications as discussed by Chen et al. [2]. The Variable Block Size Motion Estimation (VBSME) feature plays a vital role in the H.264 video coding standard. The input video sequence is processed as frames and each frame is divided into macroblocks. As the Human Visual System (HVS) is more sensitive to luma than the chroma components, the H.264 standard uses 4:2:0 sampling format as mentioned by Richardson [8], in which each macroblock consists of one block of $16 \times 16$ pixels for representing the luminance component and two blocks of $8 \times 8$ pixels for representing the chroma components as shown in Fig. 1.

The $16 \times 16$ block known as MacroBlock (MB) is the basic building block of the video coding standard which is subdivided into $16 \times 8$ or $8 \times 16$ or $8 \times 8$ sub-blocks. An $8 \times 8$ sub-macroblock

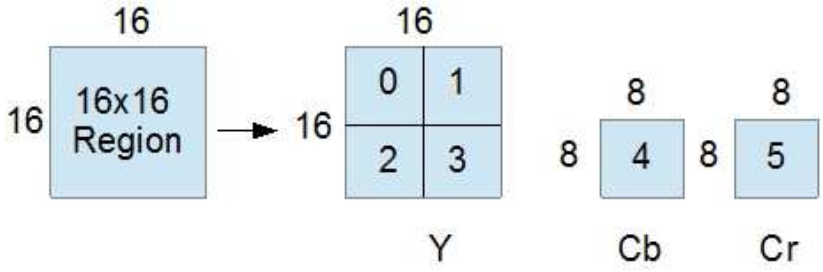



Figure 1: Macroblock (4:2:0)

can be further divided into $4 \times 4$ or $8 \times 4$ or $4 \times 8$ blocks. According to Wen et al. [9], for objects with different motions and directions, the VBSME feature has improved the coding performance in H.264 standard than the FBSME method adopted in the earlier standards. This feature has enabled the motion estimation algorithms in the H.264 standard to find its applications in video conferencing, video streaming, surveillance, video telephony, multimedia networks, etc as mentioned in Li & Yang [7].

### 1.1 Concepts involved in block matching algorithms

In the encoder design, the ME process occupies about 53% of the time when one reference frame is used and about 70.20% of the total encoding time, when four reference frames are used, as mentioned in Huayi et al. [5]. It increases the overall computational complexity in the encoder design. The probability of MV distribution has to be analyzed to obtain better match accuracy with less number of search points in the ME algorithms. In a video sequence, more than 70% of the blocks are considered as either stationary at the location (0,0) or quasi stationary at the location (±1,0) or (0,±1). As discussed by Zhao & Xu [10], in the $5 \times 5$ search area as shown in Fig. 2, the higher possibility of MV occurs at the centre position A, which is about 45.49%.

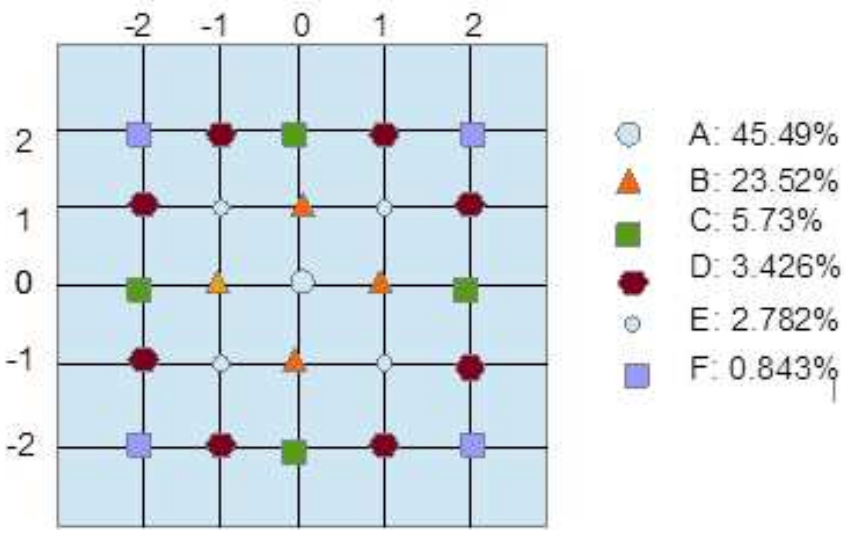



Figure 2: Probability of MV distribution in the $5 \times 5$ search area

At a pixel distance of ±1 from the centre, the MV is distributed for about 23.52% which is indicated as position B. At a pixel distance of ±2 from the centre, the possibility of distribution is about 5.73% indicated by position C. At the positions D, E and F, the possibility of MV distribution is very less and the sum of probability is about 7.051%.

### 1.2 Limitations of the existing algorithms

The Diamond Search (DS) algorithm proposed by Zhu & Ma [12] starts searching with Large Diamond Search Pattern (LDSP) and this process is repeated a number of times until the best match is obtained. Then final refinement is done using Small Diamond Search Pattern (SDSP) which is applied only once. Zhu et al. [11] proposed Hexagon Search (HS) algorithm that achieves better speed improvement than the existing DS algorithm with similar distortion performance. It is used for implementing the video codec standards such as H.261, MPEG-1 and MPEG-2. Cheung & Po [3] proposed Cross-Diamond-Hexagonal Search (CDHS) algorithm which uses cross shaped pattern followed by hexagon search pattern. It performs faster searching than DS algorithm. The New Cross Hexagon Search (NCHEXS) hybrid algorithm proposed by Belloulta et al. [1] is developed for H.264 standard. It utilizes hybrid search patterns such as the cross search and the hexagon search patterns for search path analysis.

The performance of Motion Estimation algorithms are influenced by the number of search points for many video based applications. The traditional ME block matching algorithms used in the earlier standards are based on fixed block size and are not able to incorporate different motions in the video frames. For real time video sequences with different motions, the H.264 standard adopts variable block size for video processing to address the complexity problems. The algorithms mentioned in the literature adopt fixed and hybrid search patterns which have reduced the complexity in ME Time. It is necessary to develop the motion estimation algorithm based on motion vector characteristics with quality improvement. The objective of the paper is to propose new efficient ME algorithm for improving the search efficiency using hybrid search patterns and to have less time consumption.

The rest of the section is organised as follows: Section 2 explains about the proposed algorithm and its search path analysis. In section 3, the performance analysis of the proposed and existing algorithm is done using MATLAB software. Section 5 illustrates the live streaming of the proposed algorithm. Finally, section 6 concludes the future possibilities of the work.

## 2 Proposed NCDHS algorithm

The NCDHS algorithm involves both coarse grain and fine grain search to find the best matching between the successive frames. Initially, we consider the Motion Vector Point (MVP) to be at the position (0,0). The first step is designed using unsymmetrical Cross pattern with step size $\pm 1$, and followed by modified asymmetric Hexagon with step size $\pm 1$ and $\pm 2$. The last step in which the fine grain is done which utilises either a cross or diagonal Hexagon depends on the minimum cost with step size $\pm 1$ from centre point. The steps for proposed algorithm are given below:

**Step 1:** The Coarse grain search starts with the unsymmetrical Cross search pattern with pixel distance $\pm 1$ as shown in Fig. 3. The minimum cost point is obtained using SAD computation. If the minimum cost is found to be at centre point, then the search stops else the search continues to Step 2.

**Step 2:** The half hexagon search is performed with pixel distance $\pm 1$ and $\pm 2$ as shown in Fig. 4 as a second search with minimum cost point from Step 1 acting as a centre point. This reduces the number of search points as the grid chooses the minimum cost either in the upper half or lower half of the hexagon pattern.

**Step 3:** Switch the search from the coarse to the fine resolution inner search to find the best motion vector.

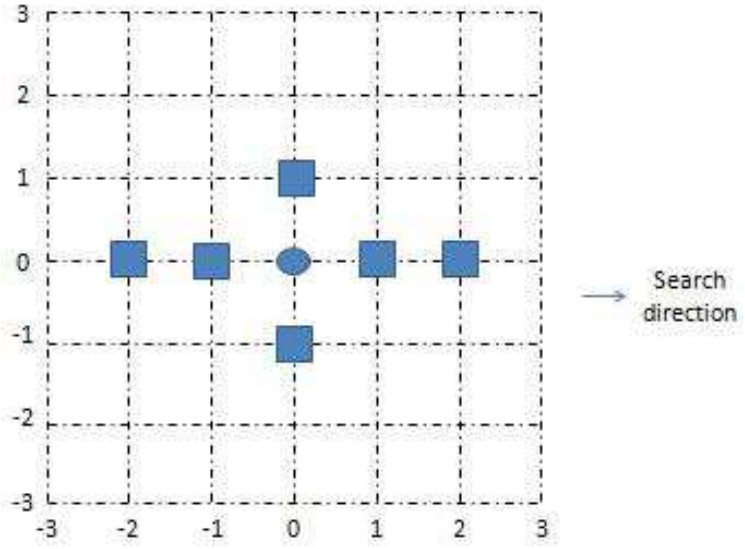



Figure 3: First search

(i) A small cross pattern with best search point obtained from global search is formed as a centre. If the current minimum SAD is at centre, the search stops. Else it moves to next step 2.

(ii) The same cross pattern is repeated based on the minimum SAD position. If the current and the previous minimum SAD positions are same, then the selected search point is the best match point else move to step 3.

(iii) Diagonal-Hexagon search pattern is formed with previous minimum SAD point as centre.

The algorithm steps are iterated until the final best vector is obtained as shown in Fig. 5.

The proposed NCDHS algorithm adopts the Pseudo code as shown in Algorithm 1 to find the minimum cost.

### 2.1 Theoretical search point analysis

Based on the assumption, let the best match block occur at the position (3,0) as shown in Fig. 6. Initially the search starts with unsymmetrical cross search pattern with seven check points. If the MBD point is at the location (1, 0), the algorithm searches for minimum motion vector using half hexagon grid with two check points (2,-1), (3,0) in addition with the pixels at the location (1,0), (0,-1) and (-1,0) that is reused from the previous search.

With the minimum MBD point found at location (3,0) from second search, the NCDHS algorithm finally searches for four check points around it, to form a small cross pattern. The search point located at (3,0) is found by using 12 search points by adopting NCDHS algorithm.

## 3 Performance analysis

To evaluate the performance of the NCDHS algorithm in terms of ME Time, search points and PSNR, simulation is done using MATLAB for the proposed and the existing algorithms at a frame rate of 30 fps. The test sequences used are "Foreman" and "Mobile" with a resolution

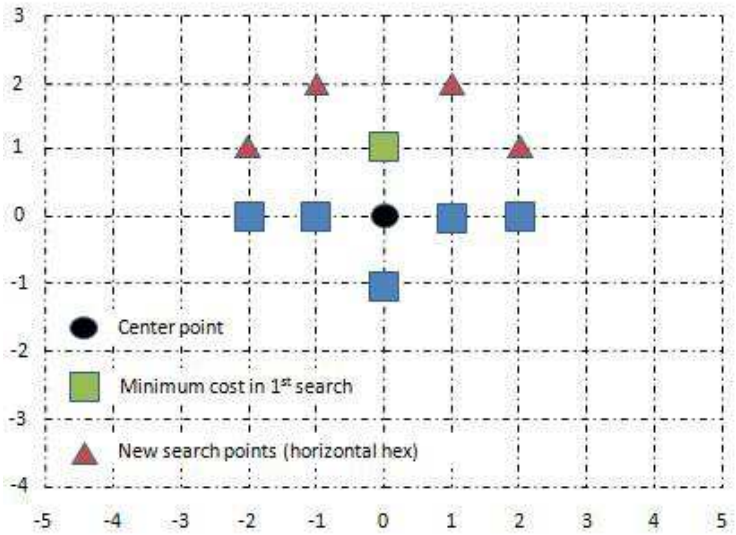



Figure 4: Second search

of 352 × 288, "Bqmall" and "Mobisode" with a resolution of 832 × 480, "People on street" and "Traffic" with a resolution of 2560 × 1600. The block size is chosen as 16 × 16 with Sum of Absolute Differences (SAD) as the distortion metric. The SAD metric at the search location $(m, n)$ is calculated as given in Eq. (1).

$$SAD(m,n) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |I_1(x,y) - I_2(x+m, y+n)| \tag{1}$$

where $N \times N$ denotes block size, $I_1(x, y)$ indicates the pixel intensity at the location $(x, y)$ in the current frame, $I_2(x, y)$ indicates the pixel intensity at the location $(x, y)$ in the reference frame and $(m, n)$ denotes the displacement vector. The lowest SAD estimates the best position of prediction occurred within the search window.

### 3.1 Motion estimation time

Table 1 summarises the simulation results obtained for the ME Time taken by the proposed NCDHS and the existing algorithms. For the mobile sequence, the ME Time obtained for proposed NCDHS algorithm requires 10.161 seconds and the ME Time obtained for the existing DS algorithm proposed by Zhu & Ma [12], HS algorithm proposed by Zhu et al. [11] and NCHEXS algorithm proposed by Belloulata et al. [1] are 19.881, 19.448 and 13.321 seconds respectively. The results show that the proposed NCDHS algorithm has reduced the ME Time when compared to the other existing algorithms.

### 3.2 Search points

The average search points for each block in a frame are calculated for different sequence in encoding 30 frames. Table 2 shows that the proposed NCDHS algorithm consumes the smallest number of search points compared to other algorithms. The results show that the average number of search points in NCDHS algorithm is 9.068. In order to demonstrate the performance of the proposed algorithm, the Fig. 7 gives the result of the average number of computations per block for different video sequences as the frame number increases. For Mobisode sequence, the search

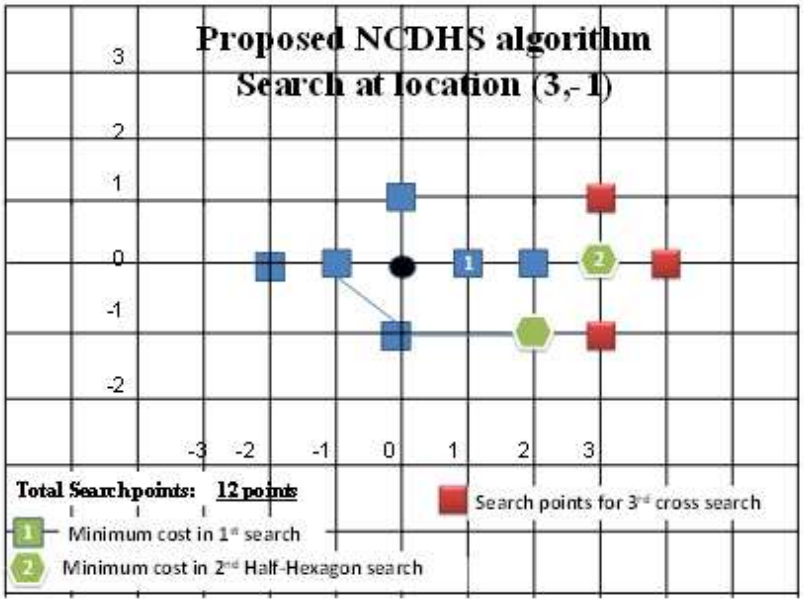



Figure 5: Search path analysis of NCDHS algorithm

Table 1: Simulation results showing ME Time of the NCDHS and the existing algorithms

| Test sequences | ME Time obtained for existing and proposed algorithms (in seconds) | | | |
|---|---|---|---|---|
| | DS [12] | HS [11] | NCHEXS [1] | Proposed NCDHS |
| Foreman | 19.387 | 19.86 | 15.233 | 9.48 |
| Mobile | 19.881 | 19.448 | 13.321 | 10.161 |
| BQmall | 61.207 | 54.765 | 48.776 | 31.506 |
| Mobisode | 77.945 | 69.422 | 57.707 | 37.461 |
| Video codec Standards | H.261, H.263 | H.261, MPEG-1,2 | H.264 | |

points obtained for NCDHS algorithm is 8.174 which is less when compared to the existing algorithms. It clearly manifests the superiority of the proposed NCDHS algorithm to the other methods in terms of number of search points used.

## 4 ME Time reduction and PSNR improvement

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 Eqs. (2)–(4):

$$\Delta\text{ME Time} = \text{MET}_{\text{proposed}} - \text{MET}_{\text{original}} \tag{2}$$

$$\%\Delta\text{ME Time} = \frac{(\text{MET}_{\text{original}} - \text{MET}_{\text{proposed}})}{\text{MET}_{\text{original}}} \times 100 \tag{3}$$

$$\Delta\text{PSNR} = \text{PSNR}_{\text{proposed}} - \text{PSNR}_{\text{original}} \tag{4}$$

**Algorithm 1** Pseudocode of NCDHS algorithm

**Input:** Initialize a large cross pattern such that $SP_H = 2 * (SP_V)$
**Output** To find minimum cost
1: **Begin**
2: **if** (MVP = minSAD) **then**
3: MV = MVP
4: **else**
5: with p = minSAD, select large half hexagon with range $\{p+2, p-2\}$ calculate the overall minSAD ($minSAD_{cg}$)
6: **end if**
7: with $minSAD_{cg}$, initialize fine-grain search with small cross with 5 points
8: **if** ($minSAD_{current} = minSAD_{prev}$) **then**
9: MV = minSAD(x,y)
10: **else if** (minSAD is in the same row or column) **then**
11: cross slides to next position
12: **else**
13: with minSAD(x,y), a diagonal-hexagon pattern is formed
14: **if** ($minSAD_{prev} = minSAD_{current}$) **then**
15: MV = minSAD(x,y)
16: **end if**
17: **end if**
18: **End**

Table 2: Average number of search points per block

| Sequence | Average number of search points (computations) for each block | | | | |
|---|---|---|---|---|---|
| | DS [12] | HS [11] | NHEXS [1] | EDS [4] | Proposed |
| Foreman | 16.839 | 12.716 | 12.084 | 12.757 | 10.48 |
| Mobile | 14.323 | 12.73 | 11.341 | 11.634 | 9.396 |
| BQMall | 15.519 | 11.439 | 11.204 | 11.307 | 8.225 |
| Mobisode | 18.793 | 13.31 | 12.982 | 13.86 | 8.174 |
| Average computations | 16.368 | 12.548 | 11.902 | 12.3895 | 9.068 |

where $MET_{proposed}$ and $PSNR_{proposed}$ denotes the ME Time and PSNR of the proposed algorithm. The $MET_{original}$ and $PSNR_{original}$ represents the ME Time and PSNR of DS, NHEXS, HS and EDS algorithms.

Table 3 shows the improvement in PSNR and the average reduction of ME Time for proposed algorithm when compared to existing DS, HS and NHEXS search patterns. It is clear that the NCDHS algorithm reduces an average of 50.11% of time for motion estimation compared to Diamond Search (DS) consequently 47.12% of Motion Estimation time compared to Hexagon Search (HS), 32.99% of Motion Estimation time compared to New Hexagon Search (NHEXS) and 43.28% of Motion Estimation time compared to Enhanced Diamond Search (EDS). This indicates that the average reduction in ME Time is greatly achieved by the proposed algorithm with slight improvement in PSNR, when compared to the excising algorithms. The comparison of the existing and the proposed algorithms is shown in Fig. 8 highlighting 30 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.

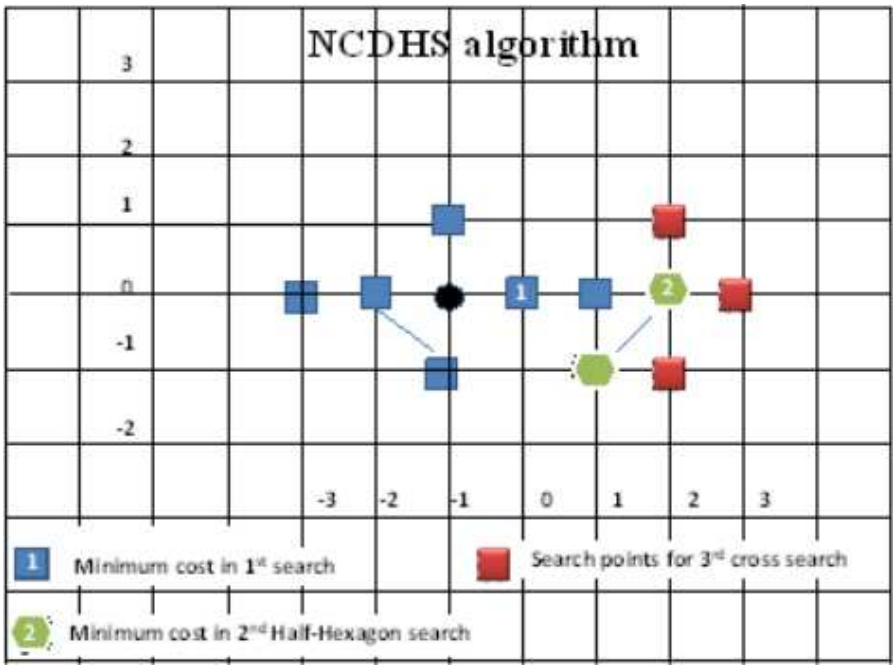



Figure 6: NCDHS algorithm search point analysis

Table 3: Simulation results showing PSNR improvement and average reduction of ME time

| Sequence | PSNR improvement (dB) (ΔPSNR) | | | | ΔME Time (in%) | | | |
|---|---|---|---|---|---|---|---|---|
| | Proposed vs DS | Proposed vs HS | Proposed vs NHEXS | Proposed vs EDS | Proposed vs DS | Proposed vs HS | Proposed vs NHEXS | Proposed vs EDS |
| Foreman | 0.57 | 0.11 | 0.06 | 0.15 | -51.10 | -52.26 | -37.76 | -51.83 |
| Mobile | 0.64 | 0.34 | 0.1 | 0.33 | -48.89 | -47.75 | -23.72 | -46.99 |
| BQMall | 0.62 | 0.24 | 0.02 | 0.05 | -48.52 | -42.47 | -35.40 | -39.07 |
| Mobisode | 0.92 | 0.84 | 0.24 | 0.59 | -51.93 | -46.03 | -35.08 | -35.23 |
| Average % time savings of the NCDHS algorithm | | | | | -50.11 | -47.12 | -32.99 | -43.28 |

## 5 Video streaming application based on NCDHS algorithm

The block diagram of video streaming system for the proposed NCDHS algorithm is shown in Fig. 9. The NCDHS server and the NCDHS client run on two MATLAB sessions on the same computer as shown in Fig. 10. After establishing the connection using Transmission Control Protocol (TCP) /Internet Protocol (IP) network, the server and client becomes ready for streaming. At the server end, the input video frames of frame size $352 \times 288$ are captured by the camera and is displayed on the PC screen. The proposed NCDHS algorithm starts estimating the motion between the frames by computing SAD values. The motion vectors are transmitted to the client over the network. The reconstruction of the video frames is done using NCDHS algorithm and the video frames are displayed on the client screen.

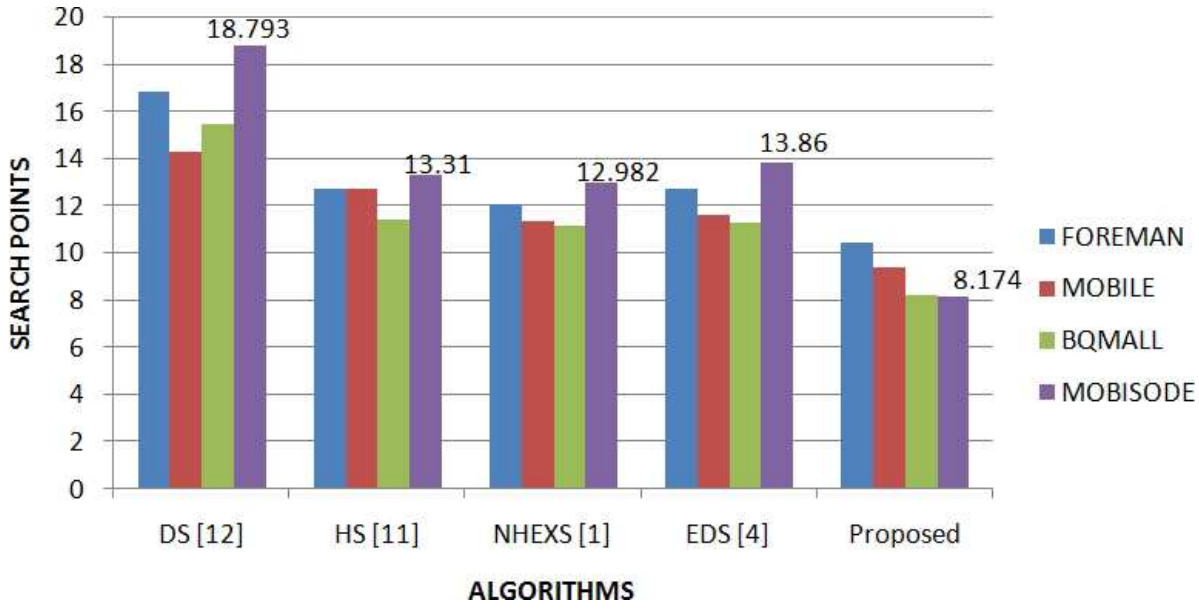
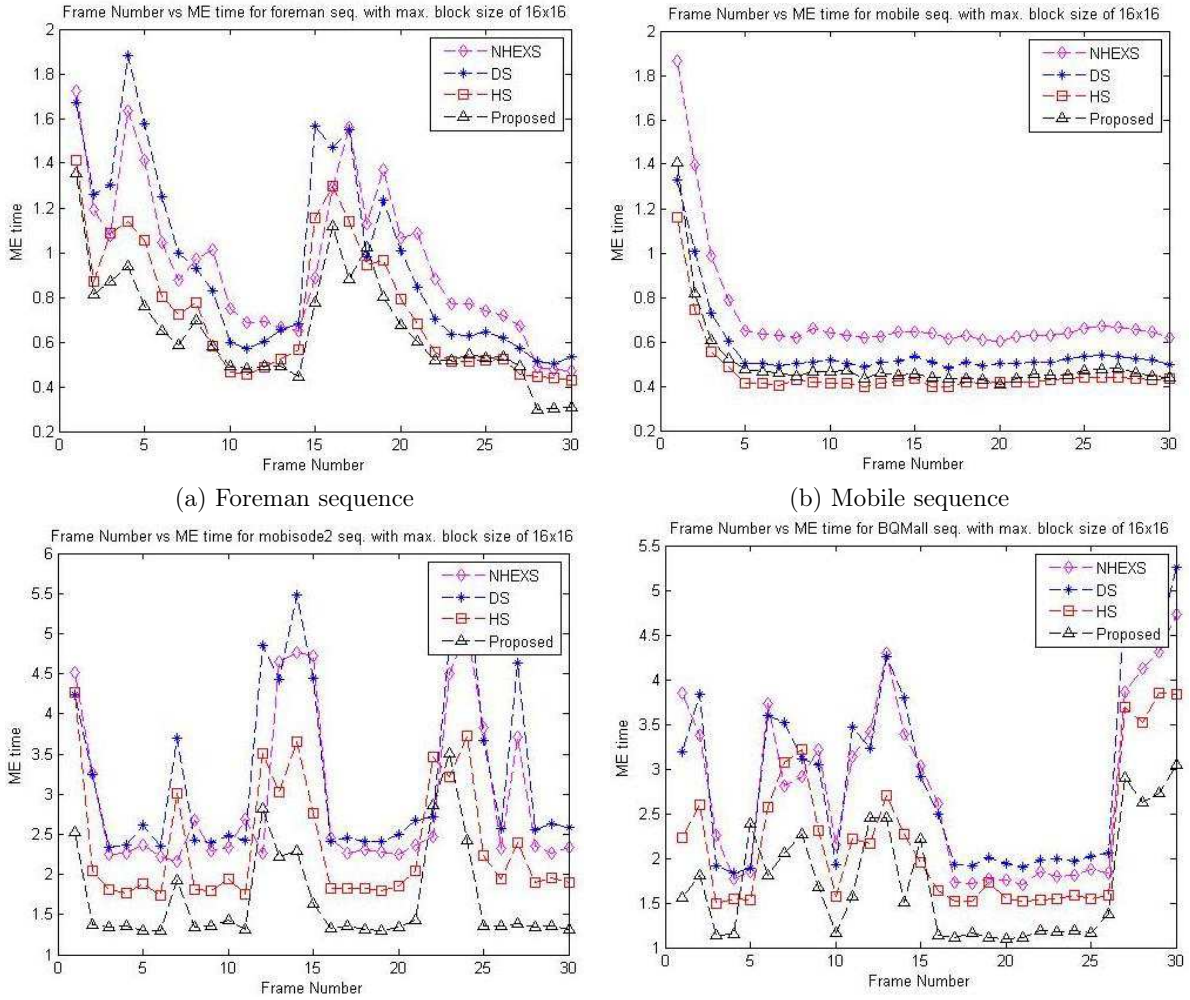



Figure 7: Performance comparison of DS, HS, NHEXS, EDS and the proposed NCDHS algorithm for (a) Foreman Sequence (b) Mobile sequence (c) Mobisode sequence (d) BQMALL sequence in terms of average number of search points




(a) Foreman sequence (b) Mobile sequence

(c) Mobisode (d) BQMall

Figure 8: Performance comparison of DS, HS, NHEXS and the proposed NCDHS for (a) Foreman sequence (b) Mobile sequence (c) Mobisode sequence (d) BQMALL sequence in terms of time reduction in ME process

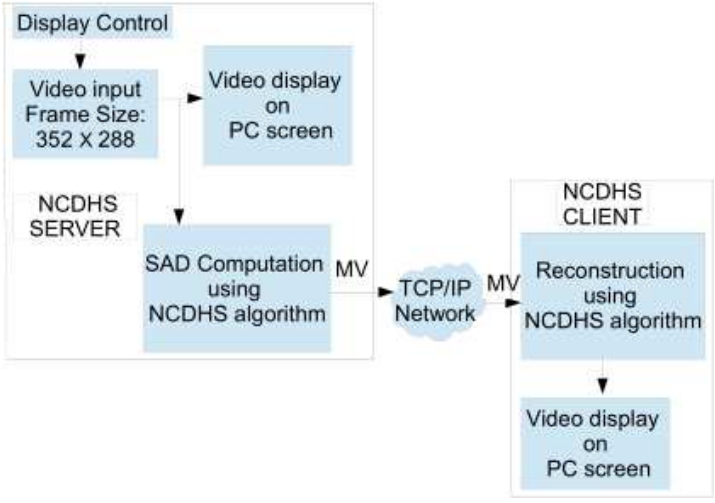



Figure 9: Video streaming using NCDHS algorithm

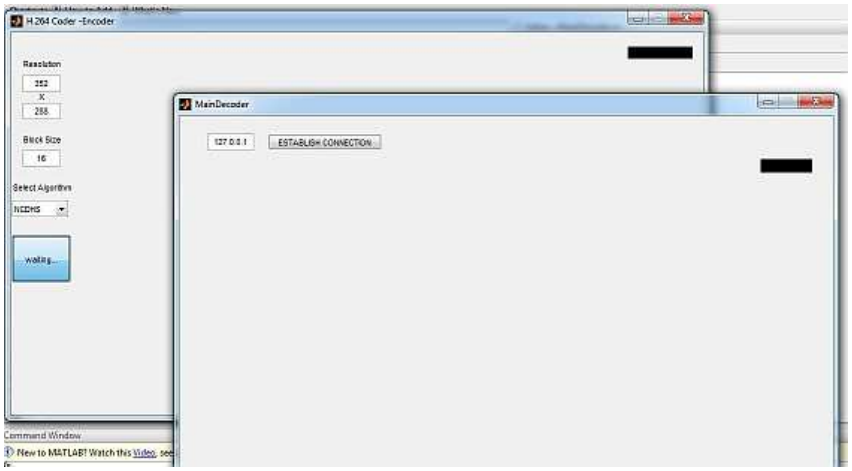



Figure 10: Snapshot of NCDHS server and client sessions using NCDHS

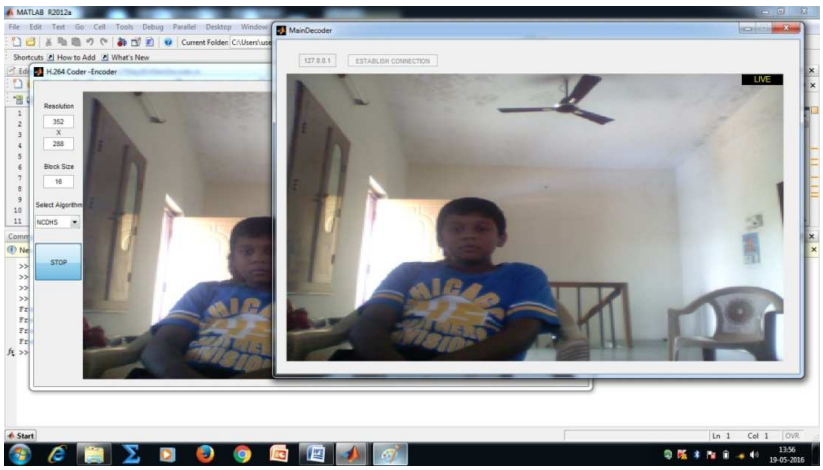



Figure 11: Sample of live input and the reconstructed output in the server and client using NCDHS algorithm

Table 4: Simulation results showing ME Time of the NCDHS and the existing algorithms

| Video Codec Standard | H.264 | | |
|---|---|---|---|
| Algorithms | NCHEXS [1] | EDS [4] | NCDHS |
| ME Time (in Seconds) | 17.18488 | 19.1599 | 14.5986 |

The sample video input and the reconstructed output is as shown in Fig. 11. For a frame size $352 \times 288$ and block size $16 \times 16$, the live video frames are captured using laptop camera. For the first thirty frames, the motion estimation time taken for the proposed NCDHS and the existing algorithms for block size $16 \times 16$ is as shown in Table 4. For the live video sequence, the ME time obtained for NCDHS algorithm is 14.5986 seconds, for the NCHEXS algorithm introduced by Belloulata et al. [1] is 19.1599 seconds and for the EDS algorithm discussed by Devi et al. [4] is 17.18488 seconds. The simulation results show that the ME time obtained by the NCDHS algorithm is less when compared to the existing algorithms. The speed of the algorithm can be further improved by reducing the motion estimation time when implemented on dedicated hardware as suitable for real time streaming applications.

## 6 Conclusion

The proposed New Cross Diagonal-Hexagon Search (NCDHS) algorithm has adopted novel half-hexagon grid global search pattern and a cross diagonal-hexagon local search pattern. The hybrid global and local search pattern utilises less search points to obtain optimal motion vector and this inturn reduces the motion estimation time, when tested on live sequences. The search points reduction has reduced the computational complexity when compared to the existing algorithms with little improvement in quality. The new block matching motion estimation algorithm using hybrid search patterns achieves less ME time when compared to the existing algorithms, which makes it suitable for live streaming application. The algorithm proposed can be extended to High Efficiency Video Coding (HEVC)/H.265 Standard by considering the reduction in motion estimation time and by extending the block size to meet the requirements of the standard.

## Bibliography

[1] Belloulata K., Zhu S., Tian J., Shen X. (2011); A novel cross hexagon search algorithm for fast block motion estimation, *International Workshop on Systems, Signal Processing and their Applications*, 1-4, 2011.

[2] Chen T.C., Lian C., Chen L.G. (2006); Hardware architecture design of an H.264/AVC video codec, *IEEE Asia and South Pacific Conference on Design Automation*, 8, 2006.

[3] Cheung C.H., Po L.M. (2005); Novel cross-diamond-hexagonal search algorithms for fast block motion estimation, *IEEE Transactions on Multimedia*, 7(1), 16-22, 2005.

[4] Devi S.R., Rangarajan P., Perinbam J.R.P., Paul R. (2013); VLSI Implementation of High Performance Optimized Architecture for Video Coding Standards, *Acta Polytechnica Hungarica*, 10(6), 237-249, 2013.

[5] Huayi L., Lini M., Hai L. (2010); Analysis and optimization of the UMHexagons algorithm in H. 264 based on SIMD, *Second International Conference on Communication Systems, Networks and Applications*, 1, 239-244, 2010.

[6] ITU-T Rec.H.264, ISO/IEC 14496-10 AVC. (2003); Joint Video Team Draft ITU-T recommendation and final draft international standard of joint video specification, 2003.

[7] Li Z., Yang Q. (2012); A fast adaptive motion estimation algorithm, *International Conference on Computer Science and Electronics Engineering*, 3, 656-660, 2012.

[8] Richardson I.E. (2011); The H. 264 advanced video compression standard, John Wiley & Sons, 2011.

[9] Wen X., Au O.C., Xu J., Fang L., Cha R., Li J. (2011); Novel RD optimized VBSME with Matching Highly Data Re-usable Hardware Architecture, *IEEE Transactions on Circuits and Systems for Video Technology*, 21(2), 206-219, 2011.

[10] Zhao W., Xu S. (2012); Research and optimization of UMHexagons algorithm based on H.264, *IEEE 4th International Conference on Multimedia Information Networking and Security*, 600-603, 2012.

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

[12] Zhu S., Ma K.K. (2000); A new diamond search algorithm for fast block-matching motion estimation, *IEEE Transactions on Image Processing*, 9(2), 287-290, 2000.