International Journal of Advanced Research in
IT and Engineering ISSN: 2278-6244

IMPLEMENTATION & COMPARATIVE ANALYSIS OF MOTION ESTIMATION ALGORITHM IN VIDEO COMPRESSION

N. K. Nakum*

Prof. A. M. Kothari**

Abstract: *In a video scene, data redundancy arises from spatial, temporal and statistical correlation between frames. These correlations are processed separately because of differences in their characteristics. Hybrid video coding architectures have been employed since the first generation of video coding standards, i.e. MPEG. MPEG consists of three main parts to reduce data redundancy from the three sources described above. Motion estimation and compensation are used to reduce temporal redundancy between successive frames in the time domain.*

Keywords: *mpeg, motion vector, search algorithms, video compression.*

*Student, master of engineering in Electronics and communication engineering at Atmiya Institute of Technology and Science, Rajkot (Gujarat-India).

**Assistant Professor, Senior Lecturer at Atmiya Institute of Technology and Science, Rajkot (Gujarat-India).

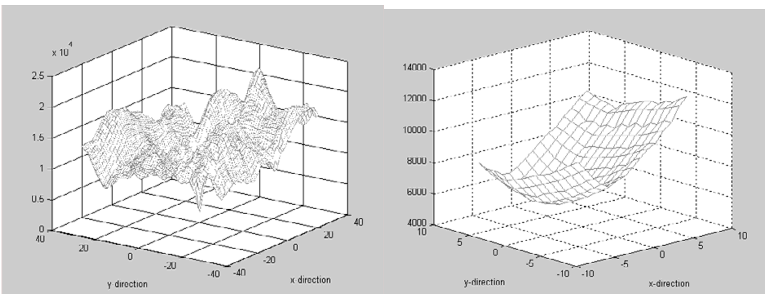
INTRODUCTION:

Digital video coding has gradually increased in importance since the 90s when MPEG- 1 first emerged. It has had large impact on video delivery, storage and presentation. Compared to analog video, video coding achieves higher data compression rates without significant loss of subjective picture quality. This eliminates the need of high band- width as required in analog video delivery. With this important characteristic, many application areas have emerged. For example, set-top box video playback using compact disk, video conferencing over IP networks, P2P video delivery, mobile TV broadcasting, etc.

The specialized nature of video applications has led to the development of video processing systems having different size, quality, performance, power consumption and cost. Similar to previous video standards such as H.261, MPEG-1, MPEG-2, H.263, and MPEG-4, H.264/AVC is also based on hybrid coding framework. Among the coding tools, Motion Estimation (ME) is the most important part in exploiting the temporal redundancy between successive frames and is also the most time consuming part in the hybrid coding framework. ME is conducted in two parts: the first part is the integer pel ME, and the second part is the fractional pel ME around the position obtained by the integer pel ME. Particularly, quarter pel motion vector precision is used in H.264/AVC to improve the coding efficiency. ME occupies 60% to 90% of computational time of the whole encoder from the simplest configuration to the complex configuration, respectively. In which the fractional pel ME occupies half of the computation time of the whole ME process.

ANALYSIS OF FRACTIONAL MOTION VECTOR

It is generally believed that the fast ME algorithm works best if the matching error surface inside the search window is unimodal.



(a) *(b)*

Fig 1 (a) Error surface of integer pel ME (b) Error surface of fractional pel ME

As shows in Fig.1, the error surface of integer pel ME is not unimodal due to the large search window and complexity of video content. So the ME search would easily be trapped into a local minimum. On the other hand, since the sub-pels are generated from the interpolation of integer pels, the correlation inside a fractional pel search window is much higher than that of the integer pel search window. Thus, the uni-modal error surface will be valid in most cases of the fractional pels. So the matching error decreases monotonically as the search point moves closer to the global minimum. In the full search method, every fractional pels around the original integer pels are treated equal. However, with the valid unimodal error surface assumption, a fast algorithm can work well if every candidate of the sub-pel refinement has different occurring probabilities. Fig 2 shows the distribution of the fractional motion vector around the best integer motion vector. It is obvious that more than 90% of fractional motion vector are at the search center in all kinds of video content. However, we still cannot just avoid the fractional part even there are huge density diagram appear near the bias search center. The small error drift of fractional part in motion vector will lead to significantly bit rate increase.

FAST BLOCK MATCHING ALGORITHMS

Exhaustive Search (ES)

In order to get the best match block in the reference frame, it is necessary to compare the current block with all the candidate blocks of the reference frames. Full search motion estimation calculates the sum absolute difference (SAD) value at each possible location in the search window. Full search computed the all candidate blocks intensive for the large search window.

Consider a block of N × N pixels from the candidates frame at the coordinate position (r, s) as shown and then consider a search window having a range ±w in both and directions in the references frame. For each of the $(2w + 1)^2$ search position (including the current row and the current column of the reference frame), the candidate block is compared with a block of size N × N pixels, according to one of the matching criteria and the best matching block, along with the motion vector is determined only after all the $(2w+1)^2$ search position are exhaustively explored.

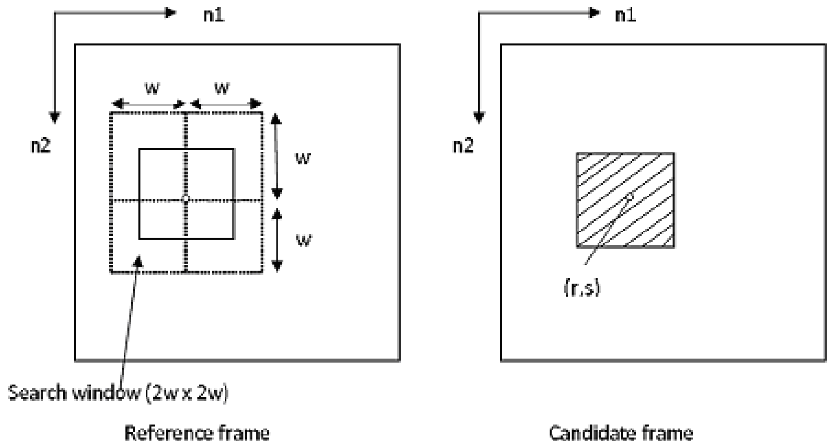


Figure-2 Full search motion estimation

Three Step Search (TSS)

This is one of the earliest attempts at fast block matching algorithms and dates back to mid 1980s. The general idea is represented in Figure 3. It starts with the search location at the center and sets the 'step size' S = 4, for a usual search parameter value of 7. It then searches at eight locations +/- S pixels around location (0,0). From these nine locations searched so far it picks the one giving least cost and makes it the new search origin. It then sets the new step size S = S/2, and repeats similar search for two more iterations until S = 1.

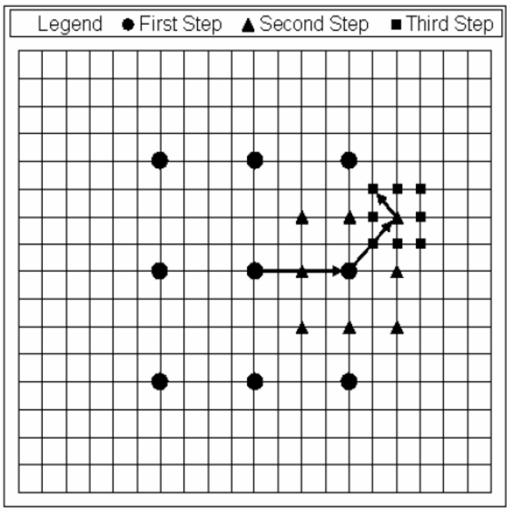


Figure- 3. Three Step Search procedure

Simple and Efficient Search (SES)

The algorithm still has three steps like TSS, but the innovation is that each step has further two phases. The search area is divided into four quadrants and the algorithm checks three locations A, B and C. A is at the origin and B and C are S = 4 locations away from A in orthogonal directions. Depending on certain weight distribution amongst the three the second phase selects few additional points. The rules for determining a search quadrant for seconds phase are as follows:

If MAD(A) ≥ MAD(B) and MAD(A) ≥MAD(C), select (b);

If MAD(A) ≥ MAD(B) and MAD(A) ≤MAD(C), select (c);

If MAD(A) < MAD(B) and MAD(A) < MAD(C), select (d);

If MAD(A) < MAD(B) and MAD(A) ≥MAD(C), select (e);

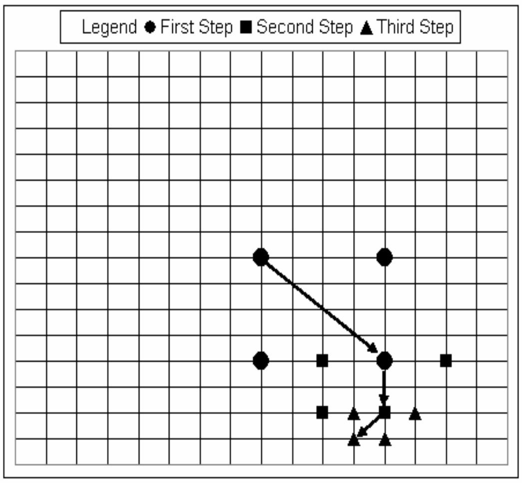


Fig. 4.The SES procedure

Four Step Search (4SS)

4SS sets a fixed pattern size of S = 2 for the first step, no matter what the search parameter p value is. Thus it looks at 9 locations in a 5x5 window. If the least weight is found at the center of search window the search jumps to fourth step. If the least weight is at one of the eight locations except the center, then we make it the search origin and move to the second step. The search window is still maintained as 5x5 pixels wide. Depending on where the least weight location was, we might end up checking weights at 3 locations or 5 locations. The patterns are shown in Fig .

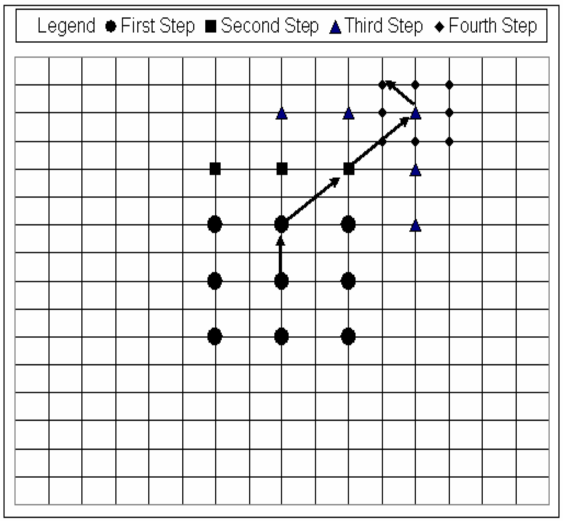


Fig.5, Four Step Search procedure

Diamond Search (DS)

The DS algorithm employs two search patterns. The first pattern, called large diamond search pattern (LDSP) shown in figure. 6 (a) comprises nine checking points from which eight points surround the center one to compose a diamond shape. The second pattern consisting of five checking points forms a small diamond shape, called small diamond search pattern (SDSP) shown in figure. 6 (b).

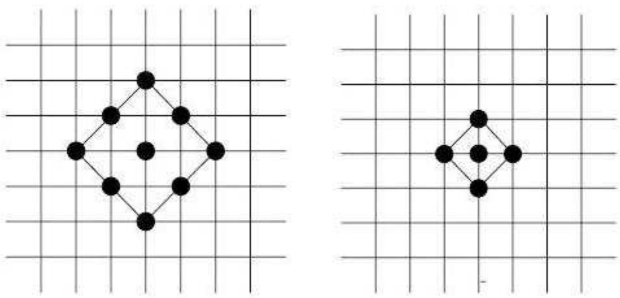
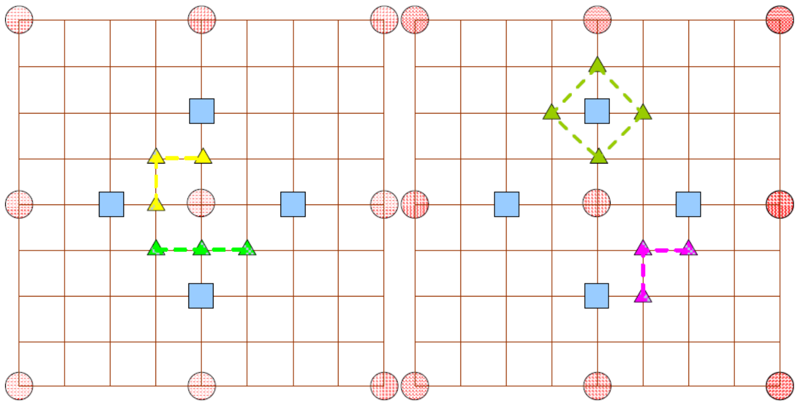


Figure.6 Diamond Search Pattern (a) Large (b) Small

Proposed Fast Algorithm

We found that the fractional motion position will be close to the center of integer search point with very high probability. To match such statistics, we will bias our second step search near the center. Moreover, with the valid uni-modal error surface assumption, we can just examine the neighborhood position around the points with low cost value and thus skips other unlikely position. We use the fixed half-pel search pattern for half pel (rectangular points in Fig 7) and adapt quarter pel search patterns. In every fractional-pel refinement, only triangle points in the same set will be visited.



(a) case 1 & 2 (b) case 3 & 4.

Figure- 7, Search pattern of proposed search algorithm (different colors-different search Pattern)

The proposed algorithm uses two step search processes, as shown in Fig 7

Step 1: Calculate the cost of five search points in half pel positions.

Step 2: Depending on the best three search positions, the search pattern of the second step is adaptively selected as shown in Fig 8 to Fig 11 The algorithm will bias the search pattern to the search center if the minimum cost point is at the integer pel, as shown in Fig 7(a). Otherwise, we will bias the search pattern away from the search center, as shown in Fig 7(b). The details of each case are shown below.

Case 1: When the minimum cost point falls on the search center, and the second and third best search positions are not neighboring to each other. We will choose the three search points between them, as shown in Fig 8.

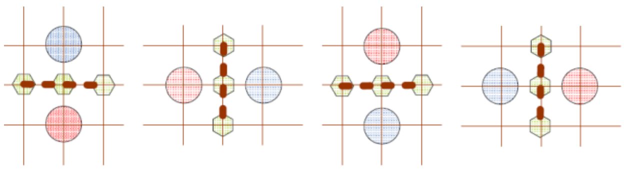


Figure- 8 Refine position in case 1 of proposed search algorithm

Case 2: When the minimum cost falls on search center and the second best search positions is neighbor to the third one. We will choose the “L” shape pattern as shown in Fig 9.

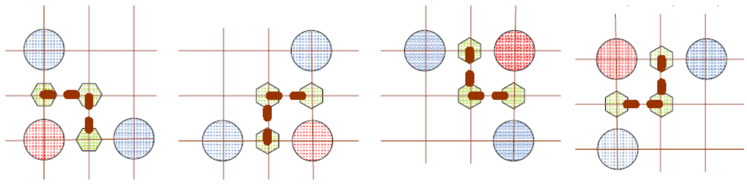


Figure- 9 Refine position in case 2 of proposed search algorithm

Case 3: When the best two search positions are at the four end points and neighboring to each other, we will search the three candidates in the "L" shape between the best two as shown in Fig 10. with search center bias.

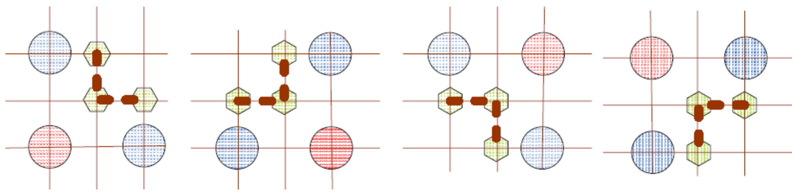


Figure- 10. Refine position in case 3 of proposed search algorithm

Case 4: When the best two search positions are at the four end points and do not neighbor to each other, we will search the four candidates around the best search point as shown in Fig 11.

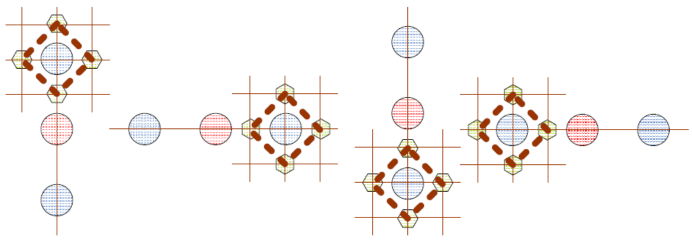


Figure- 11. Refine position in case 4 of proposed search algorithm

NKN Sequence Results

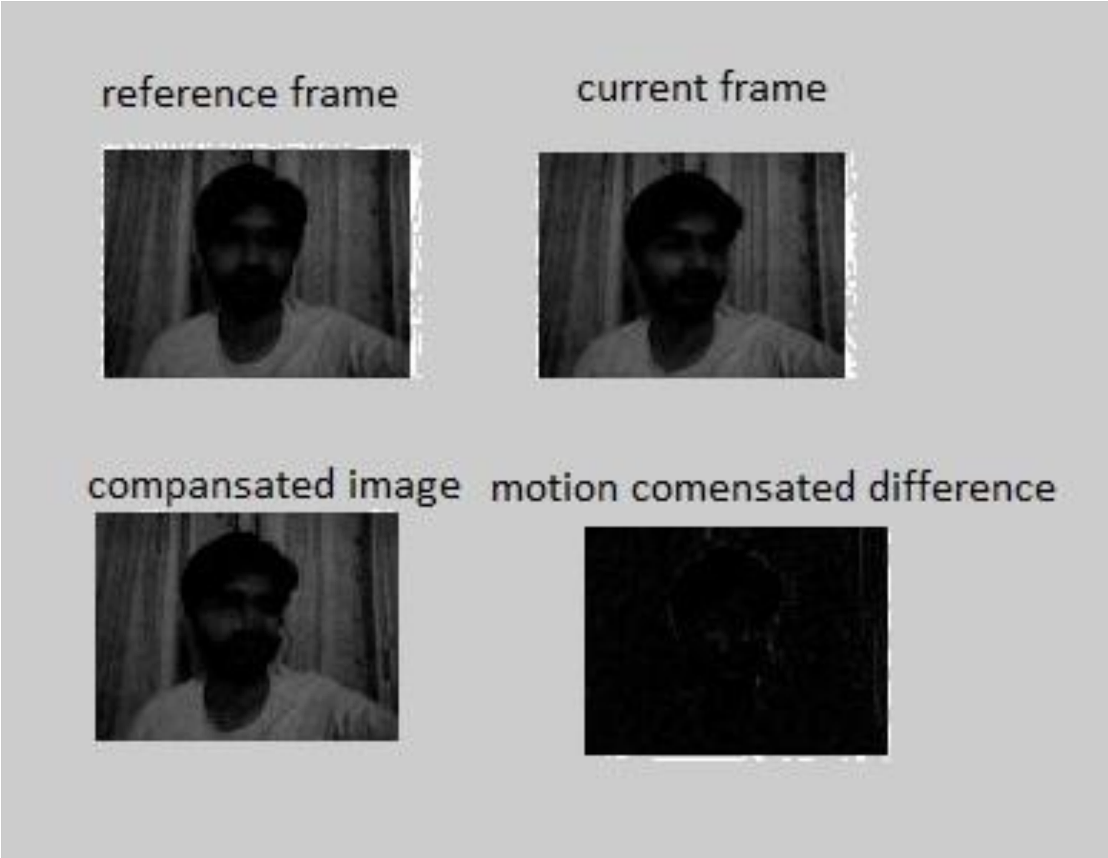


Fig.12. Compensated frame and Motion compensated difference for the NKN Sequence using proposed algorithm.

Frame Number	ES Algorithm	TSS Algorithm	SES Algorithm	4SS Algorithm	DS Algorithm	Proposed Algorithm
1	30.9113	30.7547	30.7304	30.8192	30.8923	30.8903
2	28.0086	27.9273	27.2762	27.9284	27.9639	27.9370
3	29.0486	28.7912	28.3995	28.8191	28.8736	28.9366
4	29.0486	28.7912	28.3995	28.8191	28.8736	28.9366
5	29.6373	29.2663	28.1156	29.4480	29.4721	29.3323
6	29.6373	29.2663	28.1156	29.4480	29.4721	29.3323
7	24.1784	24.0382	22.8988	23.8257	23.9437	23.9852

Table 13, NKN Sequence PSNR (dB) Results

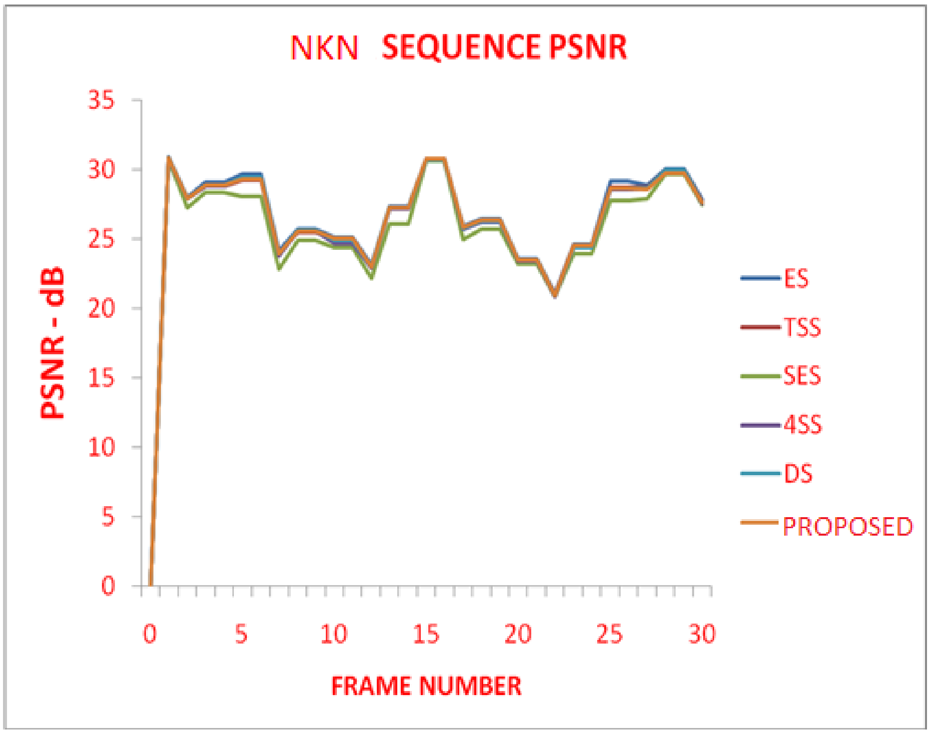


Figure-14, PSNR performance over sequence of NKN frames (PSNR performance of standard FBMAs (ES, DS, TSS, SES, FSS)) are shown in comparison to proposed algorithm

Frame Number	ES Algorithm	TSS Algorithm	SES Algorithm	4SS Algorithm	DS Algorithm	Proposed Algorithm
1	202.0485	23.0697	16.5909	16.6152	13.5758	8.5238
2	202.0485	23.1848	16.0909	18.8939	17.2697	8.6615
3	202.0485	23.2364	16.2394	18.6606	16.7546	8.6341
4	202.0485	23.2364	16.2394	18.6606	16.7546	8.6341
5	202.0485	23.2364	16.4485	18.4818	16.7485	8.6267
6	202.0485	23.2364	16.4485	18.4818	16.7485	8.6267
7	202.0485	23.2394	16.3667	20.0515	19.5576	8.7176

Table-15, NKN Sequence Computation Results

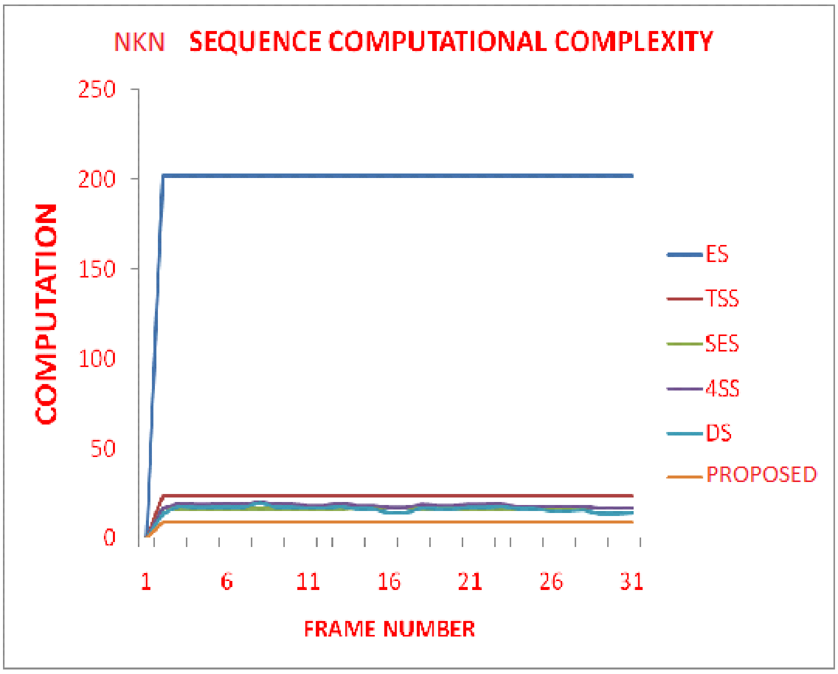


Figure- 16, Computational complexity over sequence of NKN frames (number of computations per block for standard FBMAs (ES, DS, TSS, SES, FSS) are shown in comparison to proposed algorithm)

CONCLUSION:

Because of the Internet is more and more universal and the technology of multimedia has been progressed, the communication of the image data is a part in life. In order to employ effect in a limit transmission bandwidth, to convey the most, high quality user information .It is necessary to have more advanced compression method in image and data. Motion Estimation (ME) and compensation techniques, which can eliminate temporal redundancy between adjacent frames effectively. In this algorithm we have try to improve the PSNR reduce the computation complexity.

REFERENCES:

[1] Detlev Marpe and Thomas Wiegand "The H.264/MPEG4 Advanced Video Coding Standard and its Applications", IEEE Communications Magazine, August 2006

[2] Raymond Westwater Borko Furht "Real-Time Video Compression Techniques and Algorithms", Library of Congress Cataloging-in-Publication Data, Jan 2003, pp.1-13.

[3] J.R.Jain and A.K.Jain, "Displacement measurement and its application in inter-frame image coding," IEEE Trans. Commun., Vol.COM-29,PP. 1799–1808, Dec.1981.

[4] S.kappagantula and K.R.Rao, “Motion compensated interframe image Prediction”, IEEE Trans. Commun.vol,COM 33,PP, 1011–l015,Sept, 1985.

[5] VCEG, Joint Video Team of ISO IES MPEG and ITU-T. Joint Committee Draft (CD) of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC). Wiegand, T. 2002.

[6] Farzin Ahmadianpour and M. Omair Ahmad” A Fast Algorithm for Motion Estimation under the Varying Inter-Frame Brightness Characteristics”, ICGST-GVIP Journal, Volume 5, Issue1, December 2004

[7]Michael Smith and John Villasenor ,”Intra-frame JPEG-2000 vs. Inter-frame Compression Comparison: SMPTE Technical Conference and Exhibition October 20-23, 2004, Pasadena, California

[8] Karunakar A K and Manohara Pai M M, “Motion Prediction and Motion Vector Cost Reduction during Fast Block Motion Estimation in MCTF”, International Journal of Electrical, Computer, and Systems Engineering, 2010.

[9] R. Li, B. Zeng, and M. L. Liou, “A new three-step search algorithm for block motion estimation,” IEEE Trans. Circuits Syst. Video Technol., vol. 4, pp. 438442, Aug. 1994.

[10]S.Zhu and K.-K. Ma, “A new diamond search algorithm for fast blockmatchingmotion estimation,” in Proc. 1997 Int. Conf. Information Communication and Signal Processing (ICICS), vol. 1, Sept. 9-12, 1997,pp.292-296.

[11]M. Ghanbari, “The cross-search algorithm for motion estimation,” IEEE Trans.Commun., vol. 38, pp. 950953, July 1990.

[12] L. M. Po and W. C. Ma, “A novel four-step search algorithm for fast block motion estimation,”IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313317, June 1996.

[13]E. M. Fakhouri. “Variable block-size motion estimation” citeseer.ist.psu.edu/fakhouri97variable.html.

[14] T. Wieg and Ed. Pattaya. Draft ITU-T Recommendation H.264 and Draft ISO/IEC 14496-10 AVC. In JVC of ISO/IEC and ITU-T SG16/Q.6 Doc. JVT-G050, Mar. 2003.

[15] Classification-Based Adaptive Search Algorithm for Video Motion Estimation By Mahdi Asefi A thesis presented to the University of Waterloo in fulfilment of the thesis requirement for the degree of Master of Applied Science in Electrical and Computer Engineering Waterloo, Ontario, Canada, 2006