Motion Estimation for High Speed Video Encoder

Project Report

Submitted by:

Archana Kumari

Roll No. 1301013

Under The Guidance of

Dr. Pulak Mondal

Department of Electronics and Communication Engineering

Indian Institute of Information Technology Guwahati

Guwahati 781001, India

April, 2016

CONTENTS

1. Introduction

i. Video

Video is a sequence of images where each image is function of 2-dimensional space; f(x,y). Hence, video is a 3-dimesional function; f(x,y,t).

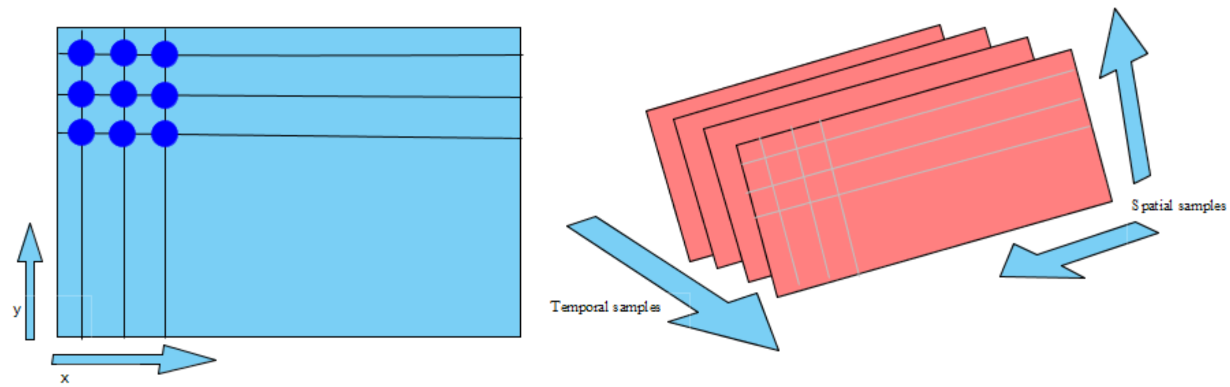


Figure 1: An image and a video sequence

ii. Video compression

Video compression means encoding raw video into fewer bits than the original representation. Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy.

Video compression is necessary because:

- Uncompressed video is of huge size which cannot be supported by many of the transmission and storage environments.
- It allows efficient usage of transmission and storage environments and even motivates to send high resolution compressed videos. Thus quality of videos enhances even after addressing to the transmission and storage issues.

iii. Video Codec

A video CODEC encodes a source image or video sequence into a compressed form and decodes this to a copy or approximation of the source sequence. Hence a video CODEC consists of an enCOder and aDECoder block A video encoder consists of three main functional units: 1. Temporal model, 2. Spatial model and 3. Entropy encoder

Temporal Model exploits the temporal redundancy between video frames and produces residual frames and a set of motion vectors describing how motion was compensated. **Spatial Model** takes the residual frame and reduces spatial redundancy among the neighbouring samples. The output of this model is a set of quantised transform coefficients. **Entropy Encoder** removes statistical redundancy in the data and produces the compressed file.

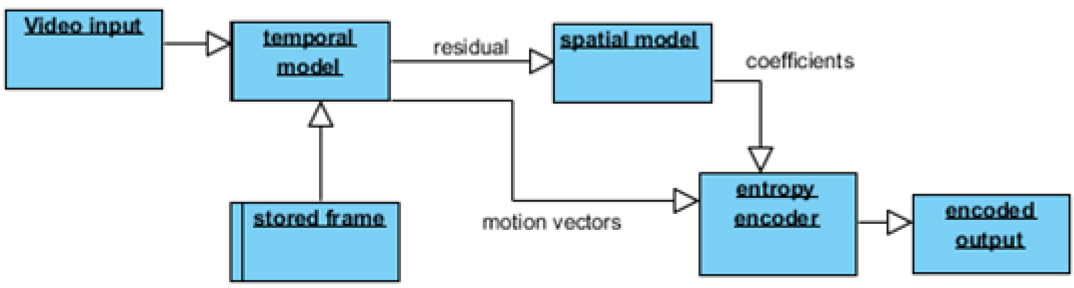


Figure 1: Video encoder block diagram

The Project work is intensely focused on the temporal model of the video encoder. The objective of this unit is to reduced temporal redundancy between transmitted frames by forming a predicted frame, corresponding to each current frame, and then subtracting the predicted frame from the current frame. This subtraction results into a residual frame which has far less energy contained in it as compared to the original frame. This residual frame is encoded and sent to the decoder where the decoder does a process reverse to it. It re-creates the predicted frame and adds the residual frame to it and gets back the original current frame.

There can be numerous ways to choose a reference frame for this operation. In the project work, previous frame is taken as the reference frame. In process of predicting a frame, inter-frame motion of pixels need to be taken into account. This kind of motion occurs due to movement of objects (rigid object motion and deformable object motion), camera motion, lighting changes, or uncovered motion, etc. There are various motion compensation methods such as block based motion estimation, sub-pixel based motion estimation, region based motion estimation, etc. The entire project work was directed with block based motion estimation.

In this method of motion compensation, for each MxN rectangular block (called as, macroblock) of the current frame i, a specified dimension of rectangular block region (called as, search area) in reference frame i-1, is searched to find a matching MxN sample region (called as, candidate block). This process of finding the best match is known as motion estimation (ME). To find the best match block, the easiest criteria is to subtract the macroblock with each search position and find the energy of the subtracted one, whichever position of search area gives minimum energy is the best match block, corresponding to which a motion vector is also generated which gives the displacement between the macroblock position and the best match position.

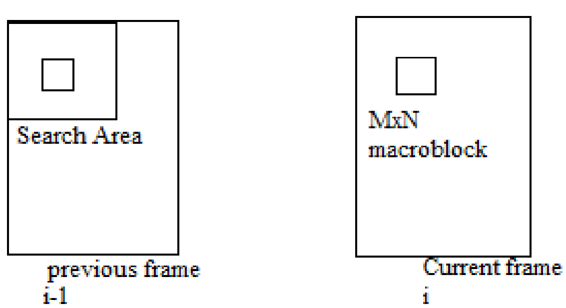
Here lies the computational complexity, of finding minimum energy to search the best match block. Again, this minimum energy can measured through numerous methods, giving rise to different parameters:

- Sum of Absolute Difference (SAD) :
  SAD = $\sum_{i=0}\sum_{j=0}|f_k(m+i,n+j) - f_{k-1}(m+x+i,n+y,j)|$
- Sum of Square Error (SSE):
  SSE = $\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(f_k(m+i,n+j) - f_{k-1}(m+x+i,n+y,j))^2$

  where MxN is the size of the macroblock, f_k and f_{k-1} are blocks of current and reference frames respectively, (m,n) is the position of the macroblock in the current frame and (x,y) is the position of block being searched. Also, there are parameters which are similar to the above two but they are averaged over total number of pixels. That is, Mean of Absolute Differences (MAD) and Mean of Square Error (MSE),such that

  $$MAD = \frac{SAD}{(M \times N)} \qquad \& \qquad MSE = \frac{SSE}{(M \times N)}$$

  Among these two parameters, MSE has more computational complexity but is more precise and results in a better search. MSE has been used in the project work.



After getting the best match block, it is subtracted with the macroblock to get the residue and a motion vector pointing the displacement between the position of macroblock and the best match position in the reference frame, is generated. Hence the motion is compensated.

In the process of block based motion estimation[1,4,9], there are many search algorithms, such as full search method (FS) [5,6], logarithmic search (LS) [8], diamond search (DS) [3], three step search (TSS) [7], Gradient Descent Search[2], etc. All these algorithms differ in their computational complexities, hence time taken in search and also in the accuracy of search, thus making the residual energies different for different search algorithms. Out of all search algorithm, FS is the most conventional and trivial algorithm for ME. This search algorithm gives the most accurate prediction but takes maximum time because of high computational complexity.

2. Objective of the project

The problem statement of the project is:

"**To design a high speed motion estimation block of a video encoder**"

The purpose of the project is to find an efficient algorithm for Block Based ME whose accuracy of prediction is higher than the already existing algorithms and time taken in search operation is also within acceptable limit.

3. Project Work

All the project works were implemented in MATLAB®2014a.

3.1. Implementation of some pre-existing ME algorithms

i. **Full Search(FS) implemented in MATLAB**

In FS algorithm, all the search positions are searched rigorously and hence it gives maximum accuracy but at the cost of time. The video frames of "foreman" were taken for this project work. In this search, search area is (16 X 16) and the macroblock has a dimension of (8 X 8).

Taking frame no. 01 as reference frame, and frame no. 02 as current frame, a predicted frame is produced in MATLAB, as shown in Figure 3 and subtracted with the current frame to get a residual frame (Figure 4).

The residual frame obtained after proper motion compensation had much less energy content as compared to that obtained after no such motion compensation,i.e. obtained by assuming the previous frame as predicted frame which means to neglect the interframe motion (Figure 4).

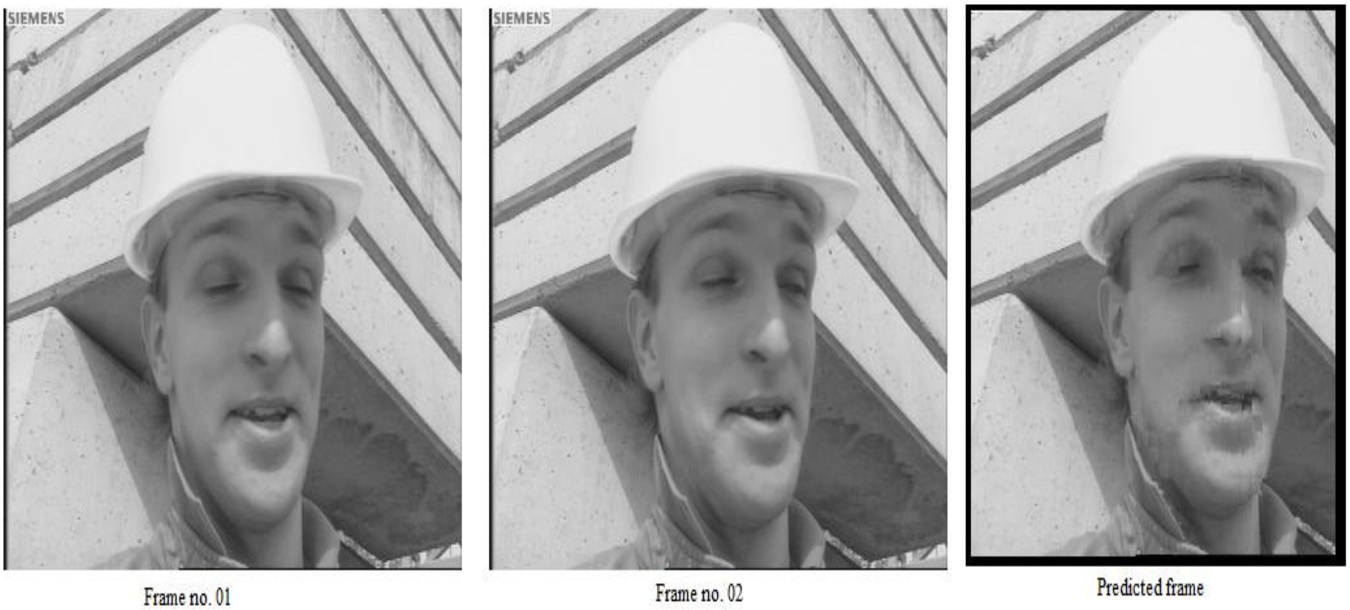


Figure 2 Original frames: Reference and current frame, Predicted frame

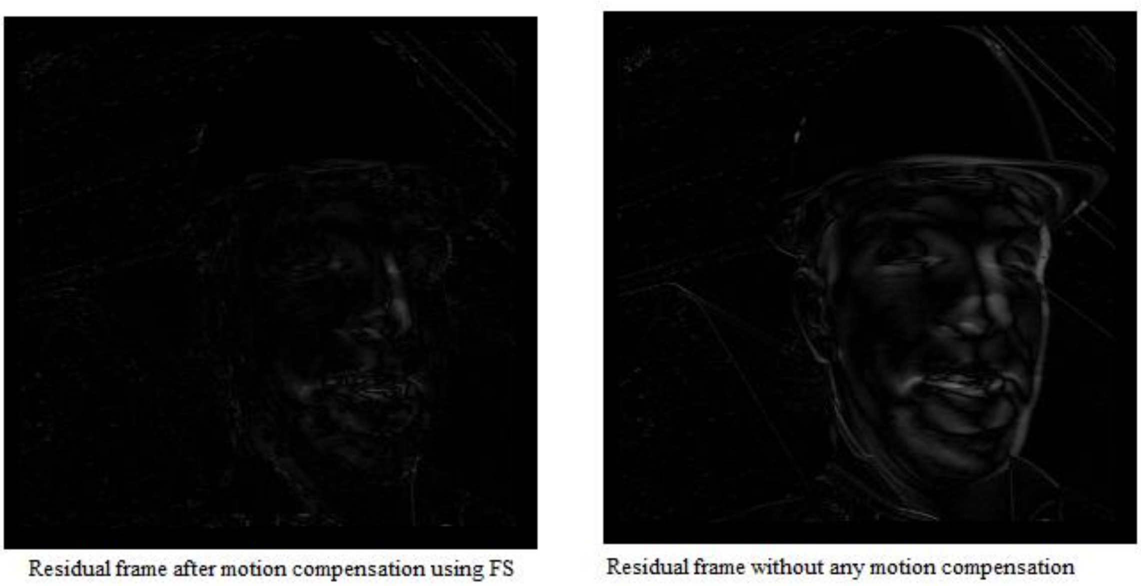


Figure 3 Residual frame with motion compensation and difference frame without motion compensation

This method was regorously applied for 10 consecutive frames of this video sequence.

It was observed that there are chances of optimization with dimension of search area and macroblocks. If search area was slightly increased or if the size of macroblocks were decreased or both, more accurate predicted frames were obtained. Again FS was implemented with search area (24x24) and same macroblock size (8x8). The results give a comparision of the both, as shown in Figure 5. Figure 5 shows that MSE values for FS applied with search area 24x24 are less than those of FS with 16x16 search area. But the time comparison states that 24x24 search area takes more time to compute the predicted frame.

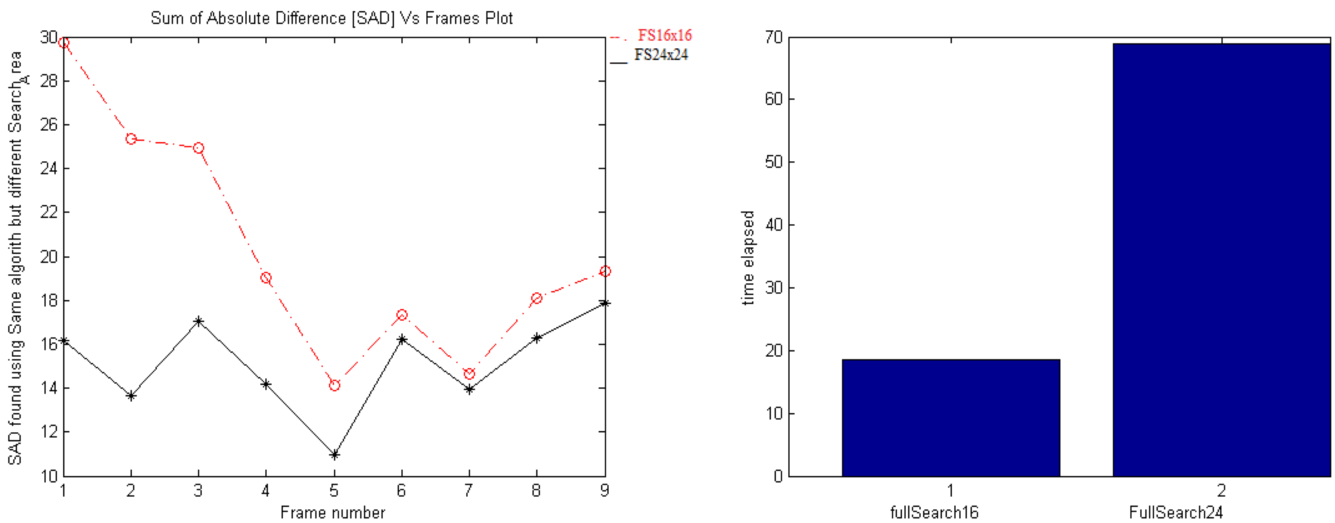


Figure 4 : Comparison of MSE values and time taken in FS using 16x16 and 24x24 search areas

Residual frame after motion compensation with 24x24 search region

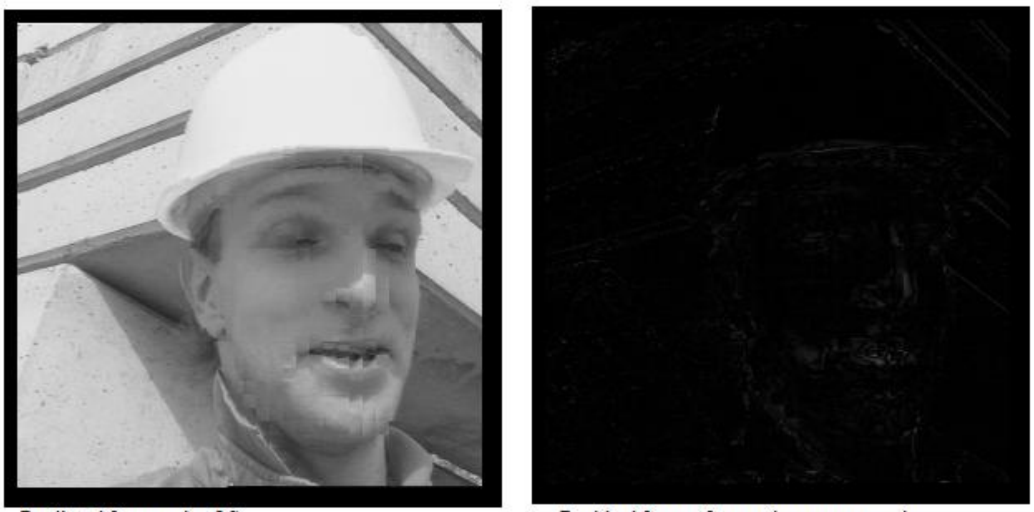
Residual frame after motion compensation with 16x16 search region

Figure 5 Residual frames obtained in FS with 24x24 search area and 16x16 search area

Figure 6 shows the difference between residual frame obtained by the same algorithm but a different search area. The residual frame obtained in FS with 24x24 search area has much less energy content in it as compared to the residual frame obtained with asearch area of 16x16. A significant improvement in prediction and motion compensation was realised by increasing the search area dimension, although it increases the computation and hence time. So it was felt better to switch to this search area dimension (24x24) instead of (16x16).

ii. Logarithmic search (LS) implemented in MATLAB:

In LS algorithm, all the search positions are searched in cross pattern. The macroblock has a dimension of (8 X 8) and the search area is 24x24. The initial step size is 4.



Predicted frame using LS   Residual frame after motion compensation

Figure 6 Predicted frame using LS and corresponding residual frame

Figure 7 shows the predicted frame obtained using LSand the corresponding residual frame. It is observed that the FS is more accurate in predicting while takes more time in computation as compared to that in LS. So, there is a trade off between acuracy and speed.

3.2. Proposed Method

An algorithm for ME is proposed here to serve the problem statement of the project:

If in a video sequence of n frames, ith frame is searched using FS and then the next three frames; (i+1), (i+2) and (i+3) frames are searched using LS, then the the average error in the prediction of frames is less than that in LS.

It is expected through this algorithm that we will be able to minimise the average or an overall error in prediction, in a video sequence. Hence to find out the actual consequences of the proposed method, it was implemented in MATLAB and the results were compared.

4. Results and Discussion:

The MSE values obtained for 10 frames for all the three implemented search algorithms show in Figure 8. Figure 8 shows that the proposed search algorithm has better performance (in terms of accuracy) than LS but FS.

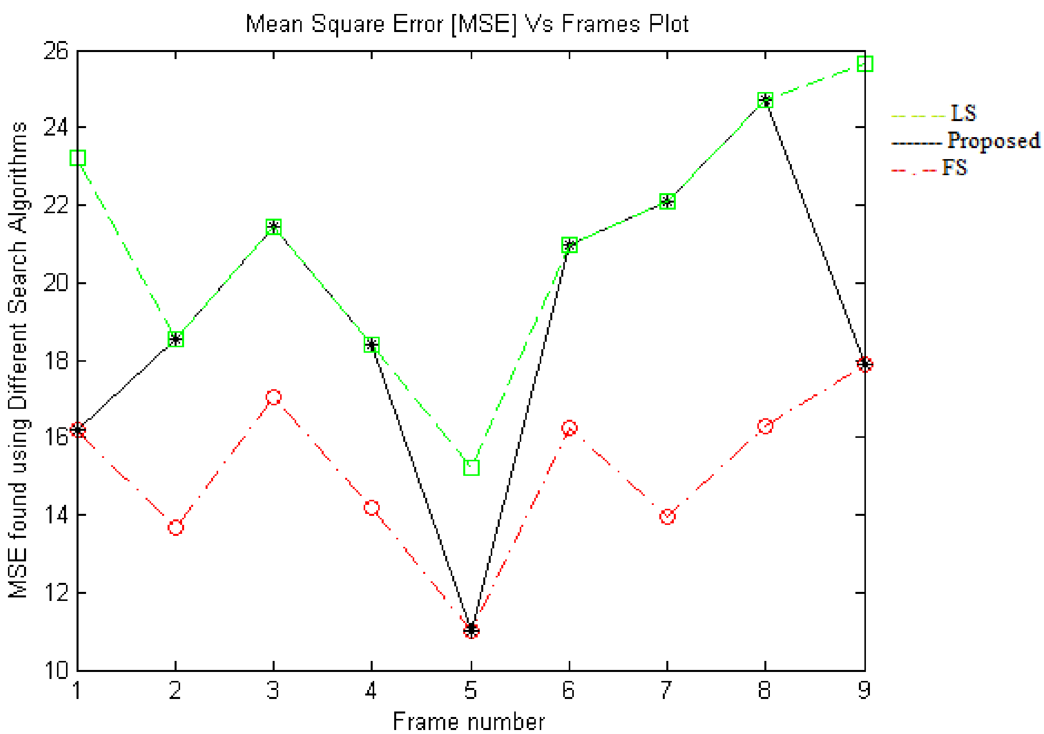


Figure 7: Comparison of MSE

The time elapsed in these three search algorithms for 10 frames were measured in the total CPU time (in seconds) used by the MATLAB® application in that particular search algorithm. Figure 9 is showing time comparison among proposed method, FS, and LS. It can be seen from Figure 9 that proposed method is much faster compared with FS but LS.

- t_elapsed_FS=67.906250000000000
- t_elapsed_proposed=31.750000000000000
- t_elapsed_LS=11.859375000000000

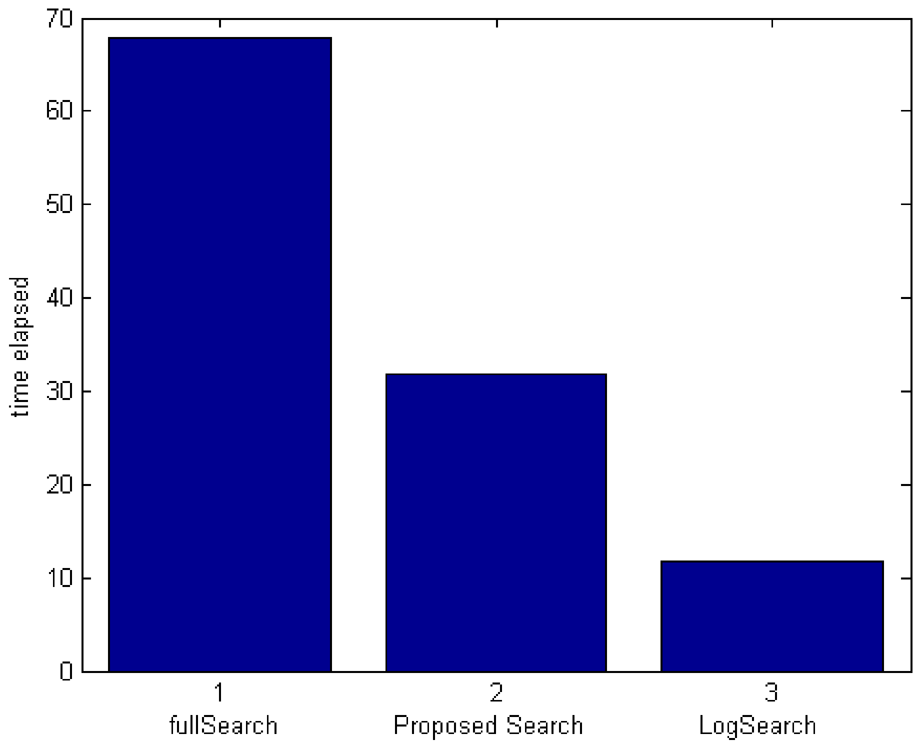


Figure 8: Time comparison

Table 1 Comparison among different ME algorithms

Algorithm	Mean Square Error for 10 frames	Time elapsed in computation for 10 frames
FS	13.639754444444446	67.906250000000000 seconds
LS	19.014694444444444	11.859375000000000 seconds
Proposed Method	17.112964444444444	31.750000000000000 seconds

5. Conclusion

Due to the presence of complex motion in between consecutive frames, many video coding algorithms are forced to reduce video quality while maintaining a target bit rate. Though FS algorithm give a good prediction, it is computationally intensive in nature. LS algorithm takes less time compared with FS but it cannot predict as accurately as FS. In this project, we have proposed a new method which exploites the better of the two algorithms.

References

1. Iain E. G. Richardson, "H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia," Wiley,2003
2. Lurng-Kuo Liu and Ephraim Feig, "A Block-Based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding," IEEE Trans. on Circuits and Systems for Video Technology, Vol. 6, No. 4, pp. 419-422, Aug. 1996.
3. Shan Zhu and Kai-Kuang Ma, " A New Diamond Search Algorithm for Fast Block Matching Motion Estimation," IEEE Trans. on Image Processing, Vol. 9, No. 2, Feb. 2000
4. Milind Phadtare, "Motion estimation techniques in video processing," Electronic Engineering Times India , Aug. 2007
5. Jong-Nam Kim and Tae-Sun Choi, Senior Member, “A Fast Full-Search Motion-Estimation Algorithm Using Representative Pixels and Adaptive Matching Scan,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, pp. 1040 – 1048, Oct 2000 .
6. Y.-C. Lin and S.-C. Tai, “Fast full-search block-matching algorithm for motion-compensated video compression,” IEEE Transactions on communications, 45(5):527-531, 1997.
7. 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. 438–442, Aug. 1994.
8. LIU, B., and ZACCARIN, “A. New fast algorithms for the estimation of block motion vectors,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 3, pp. 148 - 1572, April 1993.
9. S. Immanuel Alex Pandian, Dr.G. Josemin Bala, Becky Alma George, " A Study on Block Matching Algorithms for Motion Estimation," International Journal on Computer Science and Engineering (IJCSE), ISSN : 0975-3397, Vol. 3, No. 1, Jan. 2011