EFFICIENT VIDEO COMPRESSION SCHEMES

A thesis submitted in partial fulfilment of the requirement for the degree of

Master of Technology

In

Electronics and Communication Engineering

Specialization: Signal and Image Processing

By

Narapaneni Ragasudha

Roll No: 212EC6183

Department of Electronics and Communication Engineering,

National Institute of Technology Rourkela,

Rourkela, Odisha, 769008, India,

May 2014.

EFFICIENT VIDEO COMPRESSION SCHEMES

A thesis submitted in partial fulfilment of the requirement for the degree of

Master of Technology

In

Electronics and Communication Engineering

Specialization: Signal and Image Processing

By

Narapaneni Ragasudha

Roll No: 212EC6183

Under the Guidance of

Prof. Sukadev Meher

Department of Electronics and Communication Engineering,

National Institute of Technology Rourkela,

Rourkela, Odisha, 769008, India,

May 2014.

DEPARTMENT OF ELECRTONICS AND COMMUNICATION ENGINEERING,
NATIONAL INSTITUTE OF TECHNOLOGY,
ROURKELA, ODISHA -769008.

CERTIFICATE

This is to certify that the work done in the thesis entitled **Efficient Video Compression Schemes** by **Narapaneni Ragasudha** is a record of an original research work carried out by her in National Institute of Technology, Rourkela under my supervision and guidance during 2013-2014 in partial fulfilment for the award of the degree in Master of Technology in Electronics and Communication Engineering (Signal and Image Processing), National Institute of Technology, Rourkela.

Place: NIT Rourkela

Date: 28th May, 2014.

Dr. Sukadev Meher

Professor

DEPARTMENT OF ELECRTONICS AND COMMUNICATION ENGINEERING,
NATIONAL INSTITUTE OF TECHNOLOGY,
ROURKELA, ODISHA -769008.

DECLARATION

I certify that,

a. The work presented in this thesis is an original content of the research done by myself under the general supervision of my supervisor.
b. The project work or any part of it has not been submitted to any other institute for any degree or diploma.
c. I have followed the guidelines prescribed by the Institute in writing my thesis.
d. I have given due credit to the materials (data, theoretical analysis and text) used by me from other sources by citing them wherever I used them and given their details in the references.
e. I have given due credit to the sources (written material) used by quoting them where I used them and have cited those sources. Also their details are mentioned in the references.

N.Ragasudha

ACKNOWLEDGEMENT

This research work is one of the significant achievements in my life and is made possible because of the unending encouragement and motivation given by so many in every part of my life. It is immense pleasure to have this opportunity to express my gratitude and regards to them.

Firstly, I would like to express my gratitude and sincere thanks to **Prof. Sukadev Meher**, Head of the department, Department of Electronics and Communication Engineering for his esteemed supervision and guidance during the tenure of my project work. His invaluable advices have motivated me a lot when I feel saturated in my work. His impartial feedback in every walk of the research has made me to approach a right way in excelling the work. It would also like to thank him for providing best facilities in the department.

I would like to express my gratitude and respect to Prof. K.K.Mahaptra, Prof. S.K.Patra, Prof. A.K. Sahoo, Prof. L.P.Roy, Prof. Samit Ari, Prof. S. Maiti, Prof. A.K. Swain, Prof. D.P.Acharya for their guidance and suggestions throughout the M.Tech course. I would also like thank all the faculty members of the EC department, NIT Rourkela for their support during the tenure spent here.

I would like to express my sincere thanks to the Ph.D. scholar Mr Deepak Singh for his valuable suggestions throughout my project work which inspired me a lot. I would like to express my heartfelt wishes to my brothers, friends and classmates whose company and support made me feel much better than what I am. I would like to mention my special wishes to my juniors whose queries made my basics strong.

Lastly, I would like to express my love and heartfelt respect to my parents, sister and brothers for their consistent support, encouragement in every walk of my life without whom I would be nothing.

N.Ragasudha,

nragasudhakiran97@gmail.com

ABSTRACT

The wide use of images and videos in the day to day communication in recent trends made video compression a significant feature in information transmission and social networking. Also the limited bandwidth for transmission and limited memory make video compression a serious phenomenon to consider in the field of communication. There is a need to improve the video encoding process which can encode the video data with low computational complexity and high compression ratio along with maintaining quality quickly. The efficient encoding process is the one that is capable of removing almost all the redundancies in the video. Motion Estimation is the widely used scheme in many encoders like MPEG-2, MPEG-4 and H.264 that removes temporal redundancy. Full Search is the outdated and complex algorithm for ME. There is a wide scope of research in this particular area to reach out for the improved version of video encoder. In this thesis, many motion estimation algorithms like Full Search (FS), Logarithmic Search (LS), Three Step Search (TSS), New Three Step Search (NTSS), Four Step Search (FSS), Diamond Search (DS), Cross Diamond Search (CDS), Kite Cross Diamond Search (KCDS), Hexagonal Search (HEXS) and Enhanced Hexagonal Search (ENHEXS) are implemented and analysed. Based on the conclusions drawn from analysing the above algorithms modifications to KCDS and HEXS (Modified ENHEXS, Enhanced KCDS (ENKCDS) and Modified ENKCDS) are proposed by imparting the concept of motion vector prediction. A novel Hybrid Hexagonal Kite Cross Diamond Search (HYBHKS) algorithm is proposed. It has the capability of adaptive switching of search patterns based on the type of motion of the block that can reduce the computational complexity of the encoder maintaining acceptable quality of the video. HYBHKS performs well with less number of search points maintaining acceptable quality and high compression ratios. Analysis is done on the impartation of Discrete Hartley Transform in video encoding.

Video Compression may still be increased by reducing the information to be encoded. This is done by skipping some irrelevant information. In this thesis, compression techniques involving the concepts of spatial correlation and temporal correlation of rows are implemented. Only alternate rows of the video are fed to the video encoder. At the receiver the decoded video is resized to original dimensions by predicting the skipped rows using their neighbouring spatial and temporal rows. The technique using the spatial correlation for predicting the skipped rows perform well when compared to the one using temporal

correlation. The compression ratio and quality is high to the former scheme than that of the latter.

In this thesis, analysis is also done on down sampling and up-sampling of video before and after encoder and decoder respectively. The transmission of down sampled video makes the compression further high but with a compromise in the video quality when it is up-sampled at the receiver. Hence enhancement techniques like unsharp masking is applied to the decoded video before up-sampling it. A new technique for edge boosting using DWT is proposed in this research. It uses the HH component of DWT in edge boosting the decoded video. Then it is up-sampled. The proposed DWT based edge boosting technique gives good results when compared to unsharp masking.

INDEX

ABBREVIATIONS

DVD	: Digital Versatile Disc
HDTV	: High Definition Television
DCT	: Discrete Cosine Transform
IDCT	: Inverse Discrete Cosine Transform
DHT	: Discrete Hartley Transform
IDHT	: Inverse Discrete Hartley Transform
DWT	: Discrete Wavelet Transform
ME	: Motion Estimation
MC	: Motion Compensation
MCP	: Motion Compensated Prediction
MSE	: Mean Square Error
MAE	: Mean Absolute Error
MPEG	: Moving Pictures Experts Group
SAD	: Sum of Absolute Differences
PSNR	: Peak Signal to Noise Ratio
SSIM	: Structural Similarity Index Measure
FS	: Full Search
LS	: Logarithmic Search
TSS	: Three Step Search
NTSS	: New Three Step Search
FSS	: Four Step Search
DS	: Diamond Search
LDSP	: Large Diamond Search Pattern
SDSP	: Small Diamond Search Pattern
CDS	: Cross Diamond Search
CSP	: Cross Shaped Pattern
KCDS	: Kite Cross Diamond Search
KSP	: Kite Search Pattern
HEXS	: Hexagonal Search
HSP	: Hexagonal Search Pattern
ENHEXS	: Enhanced Hexagonal Search
EHIS	: Efficient Hexagonal Inner Search

MENHEXS : Modified Enhanced Hexagonal Search

ENKCDS : Enhanced Kite Cross Diamond Search

MENKCDS : Modified Enhanced Kite Cross Diamond Search

HYBHKS : Hybrid Hexagonal Kite Cross Diamond Search

MB : Macro-Block

CC : Computational Complexity

CR : Compression Ratio

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

THESIS OVERVIEW

1.1. Thesis Motivation

With the advent of nano technology all the electronic devices are diminished to very small sizes. The main limitation in scaling down these portable devices is the memory capacity. Hence it is necessary to use the limited memory effectively leading to the concept of data compression. Decreasing the amount of data to convey the same piece of information is known as Compression. It plays a major role in the fields of communication and signal processing in transmission and storage of data. The data can be analog or digital.

Video is defined as the collection of frames representing a continuous scene. For storing a colour video of HDTV resolution (1280×720) of 60 frames per second and of length 10s the memory requirement is

Memory Size= $1280\times720\times8\times3\times60\times10 = 1.327\times10^{10}$bits =12656 mega bits

Nearly one-fourth of a DVD of 4.7 Gb is utilized in storing this video. Also the transmission bit rate is 1266Mb/s which is very high when compared to the available HDTV channel bandwidth of nearly 20 Mb/s and highly impossible to achieve it.

The limited available memory capacity and the bandwidth motivates towards the compression of video data. It can be done at the video encoder before the transmission. The search for efficient video compression schemes dominated the research activity in video coding since early 1980s. The major milestone in video coding is H.261.

1.2. Thesis Objective

Video Compression can be achieved by exploiting the similarities or redundancies that exist in a typical video. The frames of the video are highly correlated to one another. This is called Temporal Redundancy. It can be reduced by motion estimation and motion compensation. Also each frame consists of objects of similar pixels that are connected. This is called Spatial Redundancy. This can be reduced using the transformation of spatial data to another domain where the energy is concentrated in a few coefficients.

The objective of the thesis is:

a. To analyse the video encoder and design a Motion Estimation Algorithm which can help in achieving high compression ratio with low computational complexity along with maintaining good quality of the video.
b. Study of interpolation techniques in the achievement of high compression.
c. Techniques to enhance the quality of video if the reconstructed video is of low quality.

1.3. Literature Review

Many researches have been done in the fields of image and video processing since 1980s. To achieve our objective analysis is to be done on video encoders. "**International Telecommunication Union ITU-T Recommendation H.262, 1996 Edition"** is reviewed to have knowledge on video encoding [1]. This literature describes the H.262 standard of ITU-T used in the encoding of the videos. It provides the knowledge of how the bit stream is formed by encoding the DCT coefficients of intra blocks, all the coefficients of intra and inter blocks, motion vectors, coded block patterns, headers etc. In our project, the Modified Huffman Coding tables in this standard are used for encoding the coefficients, motion vectors, coded block patterns etc. The standard Quantization matrices for intra and inter blocks are used for quantization.

Motion Estimation plays a major role in reducing temporal redundancy in videos. The paper **"Block-based Motion Estimation Algorithms- a survey" by M.Jakubowski and G.Pastuszak** presents a wide survey on the Block-based Motion Estimation Algorithms (BBME) [2]. It has described all the categories in BBME and many algorithms for implementing them. Full Search (FS), Logarithmic Search (LS), Three Step Search (TSS), New Three Step Search (NTSS) etc. are extensively described in this paper. Many new algorithms like Hexagonal Search (HEXS), Kite Search etc. are mentioned. This paper has given the basic knowledge on many fast search ME algorithms so far developed. But the patterns described in this paper cannot give low computational complexity in all cases where the motion type of the video changes. The paper on **"Enhanced Predictive Zonal Search for Single and Multiple Frame Motion Estimation"** authored by Alexis M. Tourapis [3] describes the concept of motion vector prediction in Motion Estimation which reduces the

computational complexity of the algorithm. But in this paper this concept is used for Diamond Search where the pattern consists of nine search points in the first step only which obviously increases the computational complexity predominantly in the case of estimating stationary block motion.

The paper **"JPEG Image Compression Using DHT and DCT and Comparison of Both Techniques based on Mean Square Error and Peak Signal to Noise Ratio "** authored by Dhananjey Patel, Alina Menoth Jose, Nigel Mascerenhas and Steape Stany Monis [4] describes the usage of Discret Hartley Transform in JPEG compression. It has described energy quantization and scanning order techniques that are used in encoding DHT coefficients. In this paper the compression ratio is not analysed.

The paper **"An Efficient Adaptive Unsharp Masking Based Interpolation for Video Intra frame Up-sampling"** authored by Aditya Acharya and S.Meher [5] uses the basic idea of image sharpening using unsharp masking for enhancing the quality of video. It proposed an adaptive way to choose the weight of the mask used in sharpening the video and a new interpolation technique using DCT.

1.4. Thesis Organization

The thesis is organized as follows:

Chapter 2 describes the fundamentals of image and video encoding.

Chapter 3 is focussed on the fundamental concepts of motion estimation and its role in video encoding

Chapter 4 describes the procedure of the existing motion estimation algorithms.

Chapter 5 is focussed on the proposed motion estimation algorithms in this research work and the experimental results that analyze all the discussed ME algorithms.

Chapter 6 describes Discrete Hartley Transform and its implementation in video encoding.

Chapter 7 is concentrated on the impartion of decimation and interpolation concepts in video encoding for getting high video compressions along with image sharpening techniques for enhancing the quality of the video.

Chapter 8 gives the conclusion and future scope of the research done.

CHAPTER 2

INTRODUCTION OF VIDEO ENCODING

2.1. Image Encoder

Reducing the amount of data required to represent the given quantity of information is called Compression [6]. An encoder is a device which converts the image into a coded form for compression purpose. The block diagram of a generalised image encoder is shown in fig.2.1.

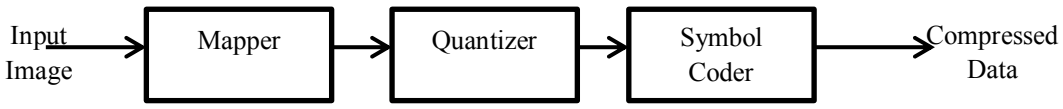

Fig.2. 1: Block Diagram of a generalised image encoder

The components of image encoder are described as follows:

2.1.1 Mapper

A mapper transforms an image into a format designed to reduce the spatial redundancy. It is a reversible process. It may or may not reduce directly the amount of data required to represent an image. For example, run length encoder yields compression in the first step of encoding. The mapping of an image into a set of less correlation transform coefficients known as Transform coding is an example of the latter. Run- length coding is advantageous only when the image is containing high spatial redundancy. Hence for real world images the latter is used.

2.1.1.1 Transform Coding

The basic objective of the transform coding is to transform data from spatial domain to the transform domain that compact the energy of the image and de-correlate the data. Discrete Cosine Transform, Discrete Hartley Transform and Discrete Wavelet Transform are the widely used transformations in image processing.

(i) Discrete Cosine Transform (DCT):-

Discrete Cosine Transform is widely used in signal and image compression schemes because of its strong energy compaction proper. A Discrete Cosine Transform (DCT)

expresses a finite sequence of data samples in terms of a sum of cosine functions oscillating at different frequencies. The two dimensional kernel of DCT is given by

$$r(x, y, u, v) = \alpha(u)\alpha(v)\cos\left(\frac{(2x+1)u\pi}{2n}\right)\cos\left(\frac{(2y+1)v\pi}{2n}\right) \quad (2.1)$$

Where $\alpha(u) = \begin{cases} \sqrt{\frac{1}{n}} & \text{for} \quad u = 0 \\ \sqrt{\frac{2}{n}} & \text{for} \quad u = 1,2,\cdots n-1 \end{cases}$ (2.2)

.*n*=row or size column of the input data,

.*x, y* are row and column wise pixel positions in spatial domain,

.*u,v* are row and column wise pixel positions in transformed domain respectively.

To reduce the computational complexity in transformation the image is portioned into blocks and DCT is performed individually to all blocks (Block Transform Coding). At the decoder the inverse takes place. The only disadvantage in Block Transform Coding is the blocking artifacts occurring at the boundaries of the block.

DCT removes the correlation among the pixels in the input data packing most of the information near to the origin i.e. (0, 0) position of the matrix. This produces many irrelevant coefficients as we move far away from the origin. These coefficients can be truncated for compression purposes.

(ii) Discrete Hartley Transform (DHT):-

Discrete Hartley Transform is similar to DCT which transforms real input to real output except that DHT expresses a finite sequence of data samples in terms of a sum of cosine and sine functions oscillating at different frequencies instead of only cosine terms [4]. The transformed coefficients of a block of pixels x(m,n) of size M×N using DHT is given by

$$X(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)\cos\left(2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right) + \sin\left(2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right) \quad (2.3)$$

Where k=0,1,2,…M-1 and l=0,1,2,..N-1.

The advantage is that the memory requirement in computation of DHT is about half of that used for DCT.

(iii) Discrete Wavelet Transform (DWT):-

DWT is generally applied to large tiles or complete images [6]. Since DCT becomes computationally intensive for larger blocks greater than 16X16 DWT is applied for this type of applications which performs better.

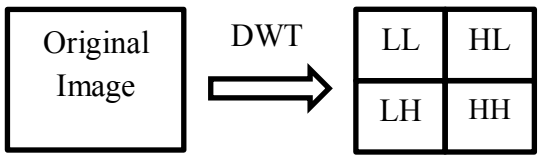

Fig.2. 2: Discrete Wavelet Transformation of an image

A single-stage DWT consists of a filtering operation that decomposes an image into four frequency bands LL, LH, HL and HH as shown in fig.2.2. LL is the original image, low-pass filtered and sub-sampled in horizontal and vertical directions. LH, HL and HH contain the residual horizontal, vertical and diagonal frequencies respectively. In multi-stage DWT, LL is further decomposed.

Most of the pixels in LH, HL and HH are zeros. Hence these can be removed and only the relevant features can be preserved for compressing the image. It performs well in the compression of still images. New JPEG-2000 uses DWT. However wavelet compression cannot be extended to temporal domain where block-based motion estimation and compensation methods are in wide use. Hence DCT is widely used in video compression techniques.

2.1.2 Quantizer

In transform coding, the information is transformed from one domain to the other without removing any information. The quantizer reduces the accuracy of mapper's output in accordance with a pre-established fidelity criterion. It is a lossy technique. It is mainly used to restrict the continuous output of transform encoder to possible levels.

Quantization has two benefits for compression.

1. If it is correctly designed, visually significant coefficients are retained while irrelevant ones are discarded.
2. A sparse matrix with a limited number of discrete levels can be efficiently compressed.

2.1.3 Symbol Coder

This is the final stage of the encoding process. Symbol coder generates a fixed or variable length code to represent the quantizer output.

(i). Fixed Length Encoder:-

In this coding, every level of the quantized output is represented by a code word of fixed length. The number of bits required to represent n levels is $\log_2(n)$. It is independent of the probability of the occurrence of pixel levels.

(ii). Variable Length Coder (VLC):-

VLC uses the probability for encoding. An efficient coding represents the less probable levels with long code and frequently occurring levels with short code.

e.g.:- Huffman Coding, Arithmetic Coding.

Since all the levels are not used in representing an image VLC is the best way of encoding.

2.1.3.1 Zig-Zag Scan

For encoding, first the two dimensional data coefficients are to be arranged in a one dimensional array. In order to scan the relevant coefficients first zigzag pattern is used. It is as shown in fig.2.3.

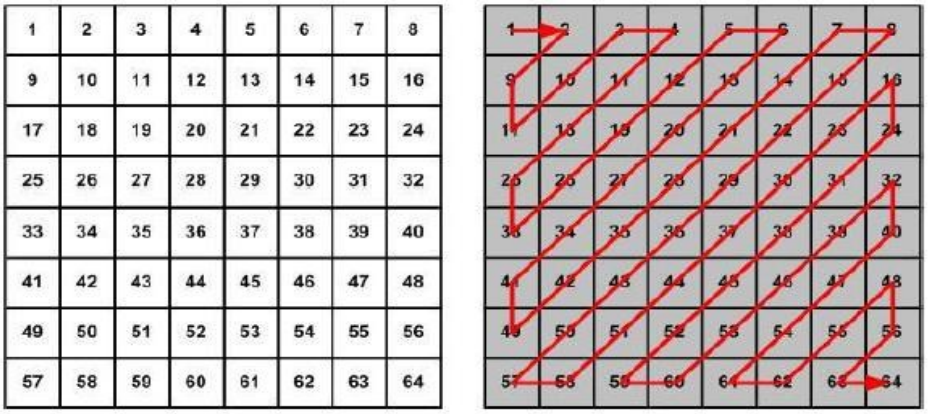

Fig.2. 3: Zig-Zag Scan Pattern of a 8X8 matrix

2.1.3.2 Run-Level Coding

The array obtained have most of the significant coefficients grouped towards the starting of the array and zeros towards the end of the array. Long sequences of identical

values can be represented as (run, level) pair where run represents the number of zeros preceding the level which is a non-zero coefficient.
Eg:- (5,12) represents five zeros preceding 12.

2.1.3.3 Entropy Encoder

The actual encoding takes place in the Entropy Encoder. The design of the entropy encoder is affected by the following constraints:

1. Compression Efficiency,
2. Computational Efficiency and
3. Error Robustness.

Huffman coding is the most popular entropy encoder. Huffman encoder uses the information of the probability of occurrences of the intensities for encoding. It designs a table of variable length codes based on the probabilities and encode the levels. The decoder should be provided with this table for decoding. In images it is an efficient scheme but in videos it adds an additional overhead for transmitting the table which reduces the compression ratio. Also it has to wait for this table till last even though the bit stream is received. Hence the Image and video standards use modified Huffman Coding cope up with these obstacles. They defined a set of code words based on the probabilities examining large video material. The decoder directly uses this table for decoding. Since it is generic a bit of compression ratio has to be sacrificed but it is more efficient since it has removed the overhead of transmitting the Huffman table.

2.2 Basic Video Encoder

A video is a collection of frames which represents a collection of frames. Encoding the frames is similar to the images [7]. Hence the design of video encoder is based on the assessments drawn from the image encoder.

2.2.1 Temporal Redundancy

The frames in the video are highly correlated with each other since all represent a same scene. Instead of sending the whole information of all the frames we can send only the information that is changing. It reduces the temporal redundancy.

A video can be encoded in two ways.

i. Intra Frame Encoding
ii. Inter Frame Encoding

i. Intra Frame Encoding:-

Intra Frame Coding refers to the various lossy and lossless compression techniques that are performed relative to the information that is contained only within the current frame, not relative to any other frame in the video sequence. In other words, no temporal processing is performed outside of the current picture of the frame.

The advantages of Intra Frame Coding are synchronization and self-decoding capability. The main disadvantage is low compression.

ii**. Inter Frame Encoding:-**

An Inter Frame is a frame which is expressed in terms of one or more neighbouring frames. Inter Frame Coding refers to the coding of inter frame. This coding is mainly used in video compression for inter frame prediction. This kind of prediction tries to take the advantage of the temporal redundancy between the neighbouring frames to attain high compression rates.

The main disadvantage of Inter Frame Coding is motion artifacts.

For decoding the inter framer the decoder requires the original previous frame. Hence the decoder uses the reconstructed previous frame which is a bit noisy. This increases the error energy. For reducing this error at the encoder the frame is reconstructed and kept in memory to use it for encoding the next frame. Since the same is used at the decoder for reconstruction it does not add any error.

2.3 Motion Estimation and Compensation

The compression ratio can be further increased if we are able to predict the current frame using the previous frame. This introduces the concept of Motion Estimation and Motion Compensation [7].

Motion Estimator estimates the motion of the object in the image and produces the motion vector.

Motion Compensator uses the motion vector and produces the motion compensated predicted frame which is more correlated to the current frame than its previous one. The difference of the current frame and the motion compensated predicted frame is sent for encoding. The motion vectors are also transmitted along with the quantization coefficients.

Including all the above blocks finally the required block diagram of basic video encoder is as shown in fig.2.4.

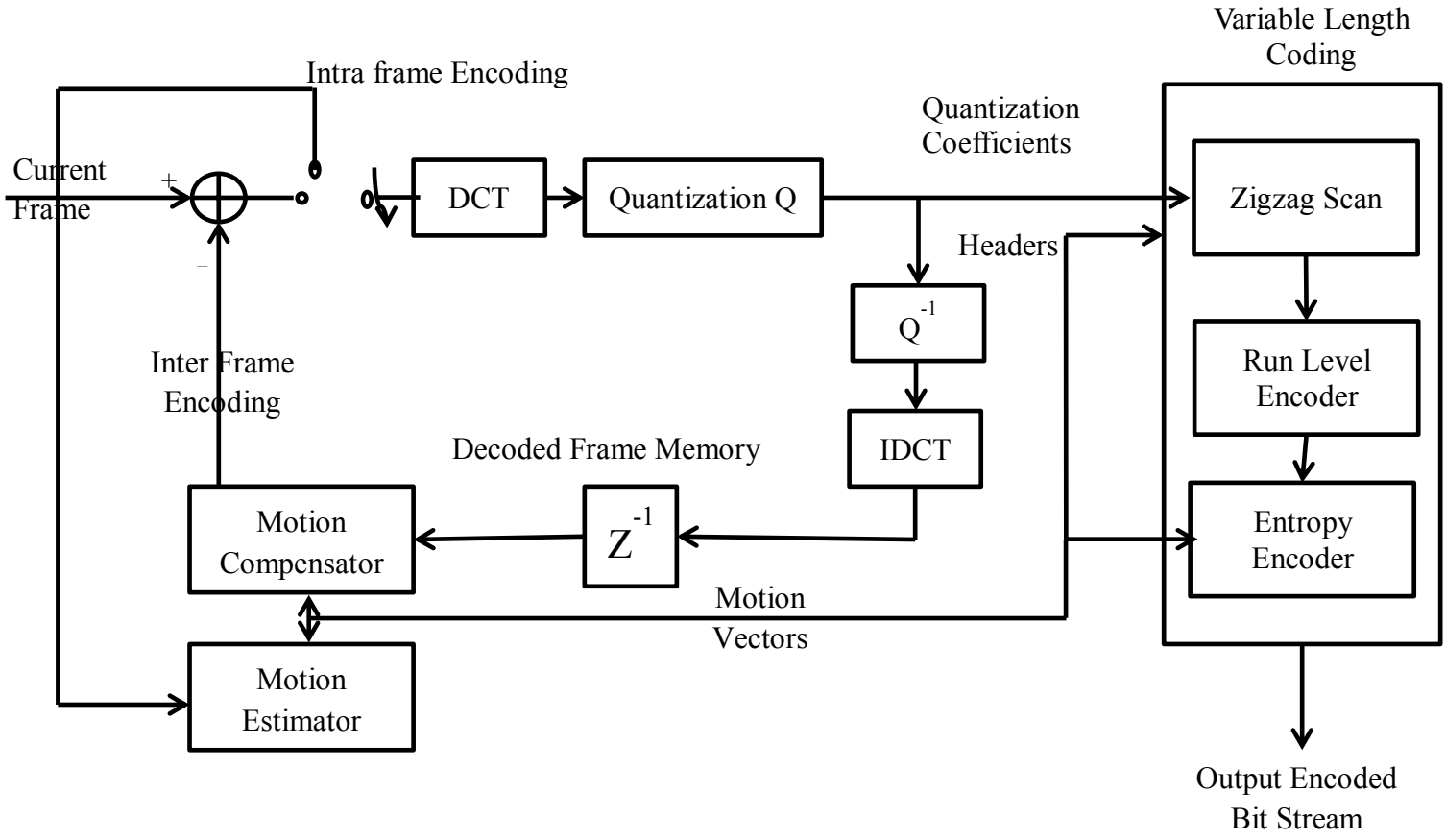

Fig. 2.4. Block Diagram of a Basic Video Encoder

CHAPTER 3

MOTION ESTIMATION

3.1 Introduction of Motion Estimation

A video is defined as the collection of frames representing a continuous scene where each frame is highly correlated to the consecutive frames. Only the moving objects in the frames change their positions while the rest of the frame remains unchanged. The difference of the current frame with the reference frame is known as Residual Frame. It consists of the information of the frame where the changes take place. The encoder defines a model describing the motion of the objects in the frame that estimate the current frame from the reference frame. This process is called Motion Estimation (ME) [7]. It then uses this motion model and moves the contents of the blocks in the reference frame to have a better prediction of the current frame. This process is known as Motion Compensation (MC) and the prediction is known as Motion Compensated Prediction (MCP). The frame thus generated is known as the motion compensated frame. The difference of the current frame and the motion compensated frame is known as MCP Residual Frame. Fig.3.1 illustrates the residual frames with and without motion estimation.

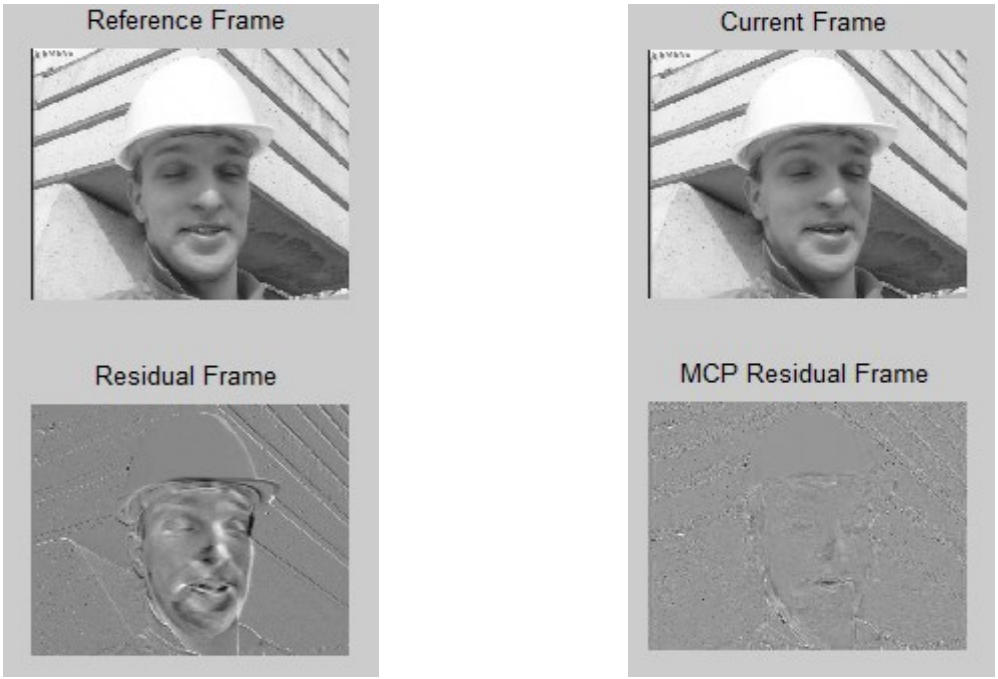

FIg.3. 1: Comparision of resiudal frames with and without Motion Estimation

The energy of the MCP residual frame is obviously less than that of the residual frame which reduces the data to represent the current frame thus increasing the compression ratio. Generally Motion Estimation is used in Inter Frame Encoding.

Motion Estimation is of two types depending on the reference frame used for prediction. It is depicted in fig.3.2.

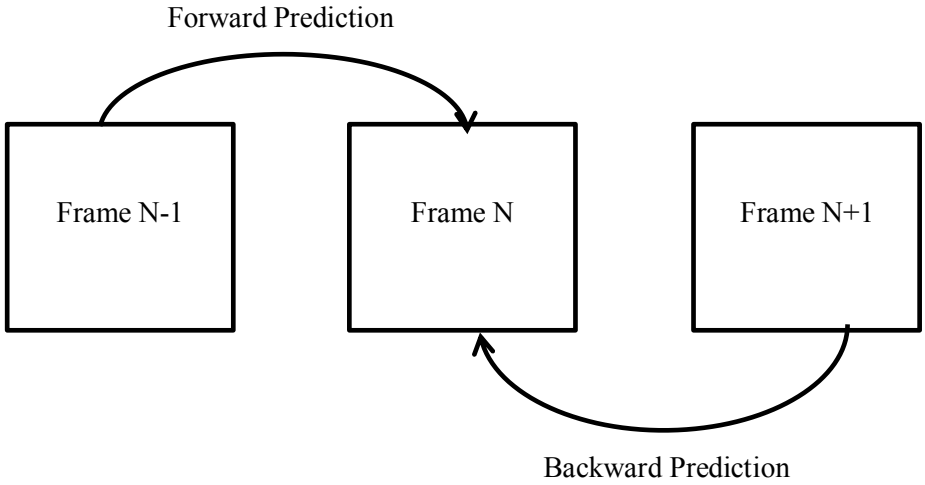

FIg.3. 2: Forward Prediction and Backward Prediction

(i). Forward Prediction:

If the reference frame occurs temporally before the current frame the prediction is known as forward prediction.

(ii). Backward Prediction:

In backward prediction, the reference frame occurs temporally after the current frame.

3.2 Block Matching Motion Estimation

Block motion compensation has become the most widely used technique for temporal redundancy removal in video coding standards. The objects in the frame will be of different sizes and of different shapes. Also only some parts of the object have the motion while the rest remains stationary. For example, consider the test video Akiyo. It has a news reporter whose lips and face move. All the other parts remain still. In such cases sending only the information about the moving areas along with the area position seems too complex to deal with. It is desirable to have a simple way to classify the objects. Here comes the role of Block Based Motion Estimation (BBME) [2]. It simply partitions a picture into blocks of size 8x8 or 16x16 and treats each block as an object and then finds its motion vector which locates the most-similar block in a prior picture (motion estimation). For each block, motion estimation

searches a neighbouring area in the reference frame for matching area. The best match is the one that has the minimum energy of the difference of the current block and the matching area. It is depicted in fig. 3.3. The search area is centered on the position of the top-left corner of the current block.

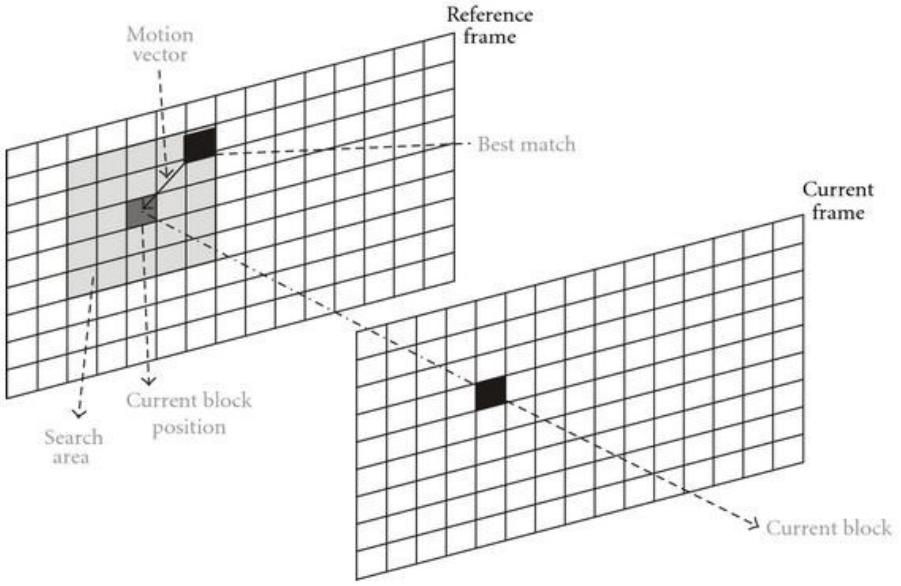

Fig.3. 3: Motion Estimation of a block

3.3 Motion Estimation Procedure:

In order to accommodate the ME technique in video encoder the following process has to be carried out at the encoder side:

1. Calculate the energy of the difference of the current block and the matching area.
2. Select the region that gives the least energy.
3. Subtract the matching region from the current block to produce a difference block.
4. Encode and transmit the difference block.
5. Encode and transmit a motion vector that indicates the position of the matching region relative to the current block position.

To reconstruct the frame at the decoder the following procedure has to be implemented:

1. Decode the difference block and motion vector.
2. Add the difference block to the matching region in the reference frame.

3.4 Motion Vectors:

A Motion Vector is defined as the relative displacement between the current candidate block and the matching block in the search window [8]. It is a directional pair representing the displacement of the block in horizontal direction and vertical direction. It can be negative also. The maximum value of the motion vector is based on the size of the search window. As the size of the search window increases the no. of search points to be checked increases which leads to increase in the computational complexity. However if the motion of the objects is fast then it is worthy to have a large search area.

3.5 Macro Blocks

In MPEG standard [1] a macro block consists of six blocks each of size 8×8 in which four blocks are from luminance part (Y) and two from corresponding chrominance part (Cb and Cr) of the luminance macro block as shown in fig. 3.4.

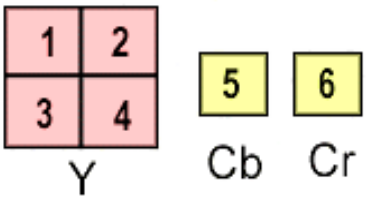

FIg.3. 4: Macro-Block

3.6 Coded Block Pattern

Most of the blocks in the inter frame will be zeros. During encoding, these blocks can be skipped. Coded Block Pattern [1] is used by the encoder to send the information regarding the skipped blocks. It is encoded for every macro block. In 4:2:0 format, each macro block contains 6 blocks (four luminance blocks and two chrominance blocks) as shown in fig.3.4. Hence the length of the CBP is fixed to 6 each representing the information whether the corresponding block is skipped or not. A '1' represents the corresponding block is not skipped where as a '0' symbolizes that the corresponding block is skipped from encoding.

3.7 Block Matching Distortion Measures

Block Matching is performed by minimizing certain block distortion measures that provide the error energy. Their mathematical formulae [7] are as follows:

(i). Mean Square Error (MSE):-

Mean Square Error is defined as

$$MSE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(C_{ij} - R_{ij}\right)^2 \qquad (3.1)$$

Where C_{ij} is a sample of the current block and R_{ij} is a sample of the reference area.

(ii). Mean Absolute Error (MAE):-

MAE is calculated as

$$MAE = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \qquad (3.2)$$

(iii). Sum of Absolute Differences (SAD):-

Sum of Absolute Differences is calculated as

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|C_{ij} - R_{ij}\right| \qquad (3.3)$$

In our project, SAD is used for calculating distortion measure between the blocks while matching because of its easy calculation.

3.8 Performance Metrics

The performance of a ME algorithm is assessed by the quality of the reconstructed video, computational complexity of the algorithm, effectiveness and encoded bitrate. The quality of the video is assessed by peak signal to noise ratio (PSNR) and Structural Similarity Index Measure (SSIM). Computational complexity is decided by the average number of search points checked per macro block. Effectiveness of the algorithm is decided by the Compression Ratio attained by it. Encoded bit rate gives the average no. of bits that are encoded in a second in order to have a specified frame rate.

PSNR can be calculated as

$$PSNR = 10\log_{10}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(\frac{255^2}{(I(i,j) - \hat{I}(i,j))^2}\right) \qquad (3.4)$$

where I (i, j) is the original frame and $\hat{I}(i, j)$ represents the reconstructed frame at the decoder. For high quality videos the PSNR should be high.

SSIM is a quality measures which compares two images [9]. SSIM of two images x and y is determined by the formula

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.5)$$

where μ=mean, σ^2=variance, $C_1=(K_1L)^2$ and $C_2=(K_2L)^2$ where L is the dynamic range of the pixel values (255 for 8-bit gray-scale images) and K_1 and K_2 are very small values close to zero($K<<1$). The greater the SSIM the better will be the quality of the image.

Compression Ratio (CR) is calculated as the ratio of size of original video file to the encoded video file given by

$$CR = \frac{Size\_of\_Original\_Video}{Size\_of\_Encoded\_Video} \quad (3.6)$$

For a compression technique to be a good one, the CR should be large.

Encoded Bit rate is the average number of bits that are encoded per second in order to encode the new video file with a specified frame rate. It is calculated in bits per second using the formula

$$Encoded\_Bit\_Rate = \frac{Size\_of\_Encoded\_video\_in\_Bits \times Frame\_Rate}{Total\_Number\_of\_Frames} \quad (3.7)$$

The lower the encoded bit rate the better will be the transmission using only a part of bandwidth of the channel. Hence much data can be sent at a time.

CHAPTER 4

MOTION ESTIMATION ALGORITHMS

The design goal of Motion Estimation Algorithm is to model the current frame as accurately as possible using previous frames or future frames whilst maintaining acceptable computational complexity. The current frame is motion compensated and subtracting the model from the frame to produce a motion compensated residual frame. This is encoded and transmitted along with motion vectors. Different Motion Estimation (ME) algorithms follow different strategies for selecting search points. Various ME algorithms are described in the following subsections.

4.1 Full Search (FS) Algorithm

Full Search is popularly known as Exhaustive Search [2]. It searches each and every search point (SP) in the search window and calculate its corresponding SAD. The search window of a current block is an area in the previous frame of maximum displacement N from the origin which is the top-left position of the current block. The position of the search point with minimum SAD is selected as the best match and the position is encoded as the required motion vector.

Computational complexity C.C. = $(2N+1)^2$

FS gives the true global minimum position in the search window. But it is computationally intensive which makes its usage limited.

4.2 Logarithmic Search (LS) Algorithm

Logarithmic Search is a multi-stage search in which the step size is reduced in each stage [7]. LS algorithm is a computationally simple algorithm. In each step, four search points in '+' shape with S step size along with the centre are searched. The flow chart of LS algorithm is as shown in fig.4.1. The algorithm is described as follows:

Step 1: Initialize S=N/4.

Step 2: Search in a '+' pattern around the origin. If minimum SAD occurs at the centre then step size is halved otherwise continue.

Step 3: Update the origin to new minimum SAD position. Repeat step 2 and step 3 till S=1.

Step 4: Eight points around the centered are checked and motion vector is obtained.

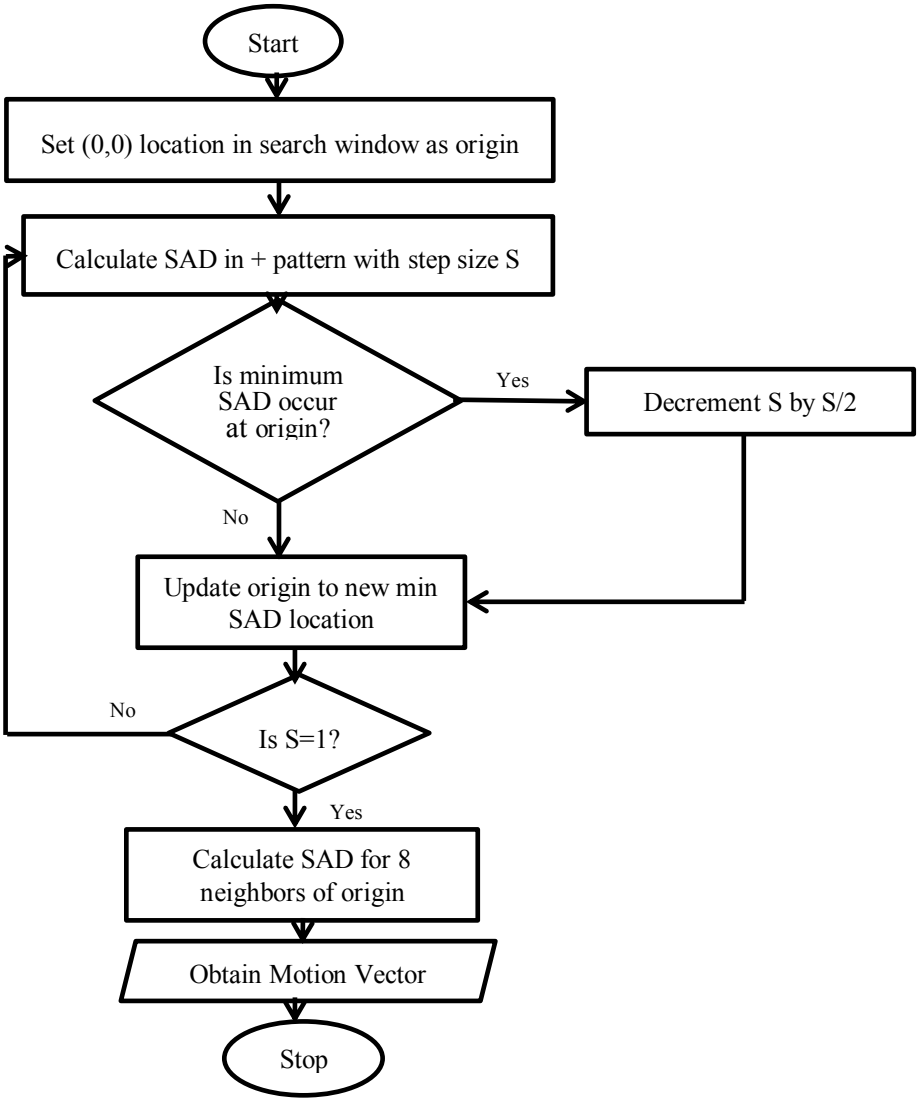

Fig.4. 1: Flow chart of LS

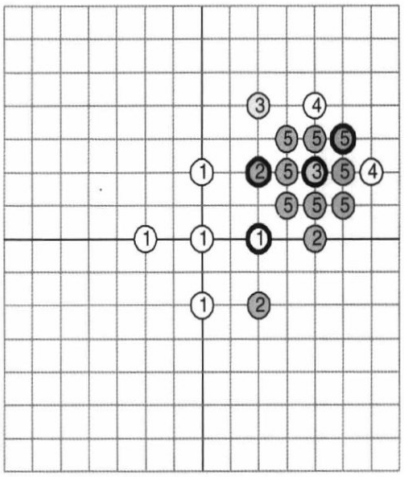

Fig.4. 2: Logarithmic Search

Logarithmic Search is illustrated as shown in fig.4.2.

$$C.C. = 5+3\times n_1+4\times n_2+8$$

Where n_1= no. of iterations in which step size is unchanged,

. n_2=no. of times step size has been reduced.

Worst case is 21 for N=7.

The disadvantage is that since very small no. of search points are checked in this algorithm true global minimum could not be obtained.

4.3 Three Step Search (TSS)

TSS uses the regular square pattern [2]. N-step search is widely used in its three step form. The flow chart is shown in fig.4.3 and illustrated in fig.4.4.

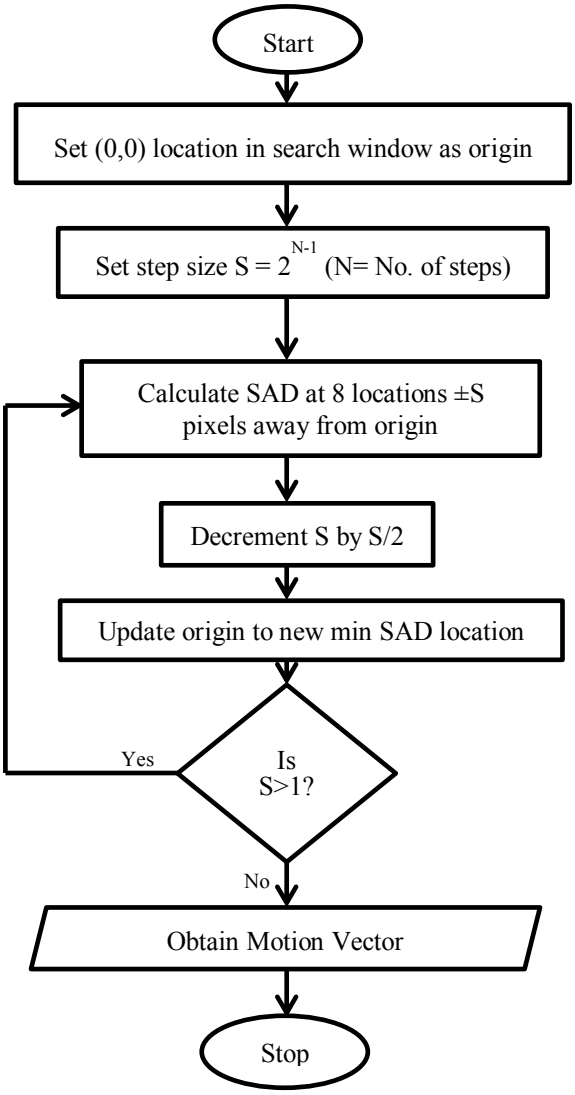

Fig.4. 3: Flow chart of TSS

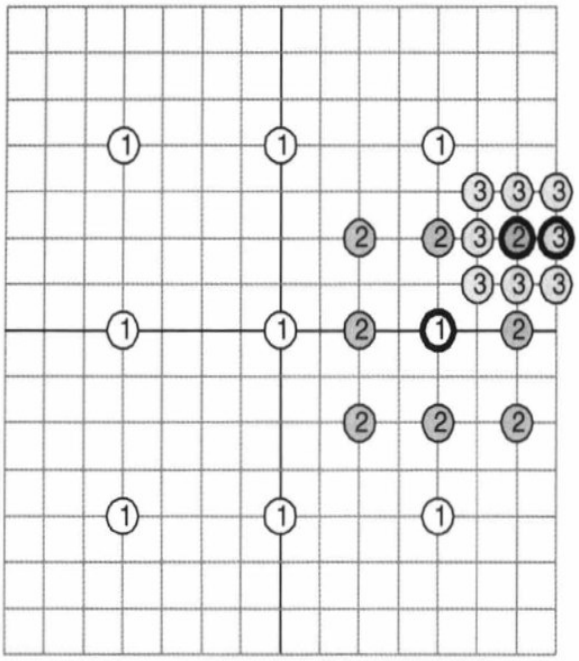

Fig.4. 4: Three Step Search

Algorithm:-

1. Initialize step size S= 2^{N-1}, where N is the no. of steps.
2. Search for eight points in ±S away from origin. Calculate minimum SAD position.
3. Update the origin to new minimum SAD point and decrease the step size by half.
4. Repeat the process for N-steps.

C.C. = $1+8\times (\log_2(N+1))$

Even though TSS is a fast algorithm it has relatively large search pattern in 1st step which renders it inefficient for finding blocks with small motions and no early terminations.

4.4 New Three Step Search (NTSS)

NTSS is an improvement over TSS for speeding up in tracking of stationary and quasi stationary blocks introducing early termination in first step [2]. In this algorithm, in first step along with eight neighbours at a distance ±S distance away from centre additional neighbours at a distance ±1 distance away from centre are searched. If the minimum SAD point occurs at the origin or any of eight neighbours ±1 distance away from centre, the search is stopped. Otherwise the centre is updated and TSS is continued. The flow chart of the algorithm is presented in fig.4.5 and the illustration is given in fig.4.6.

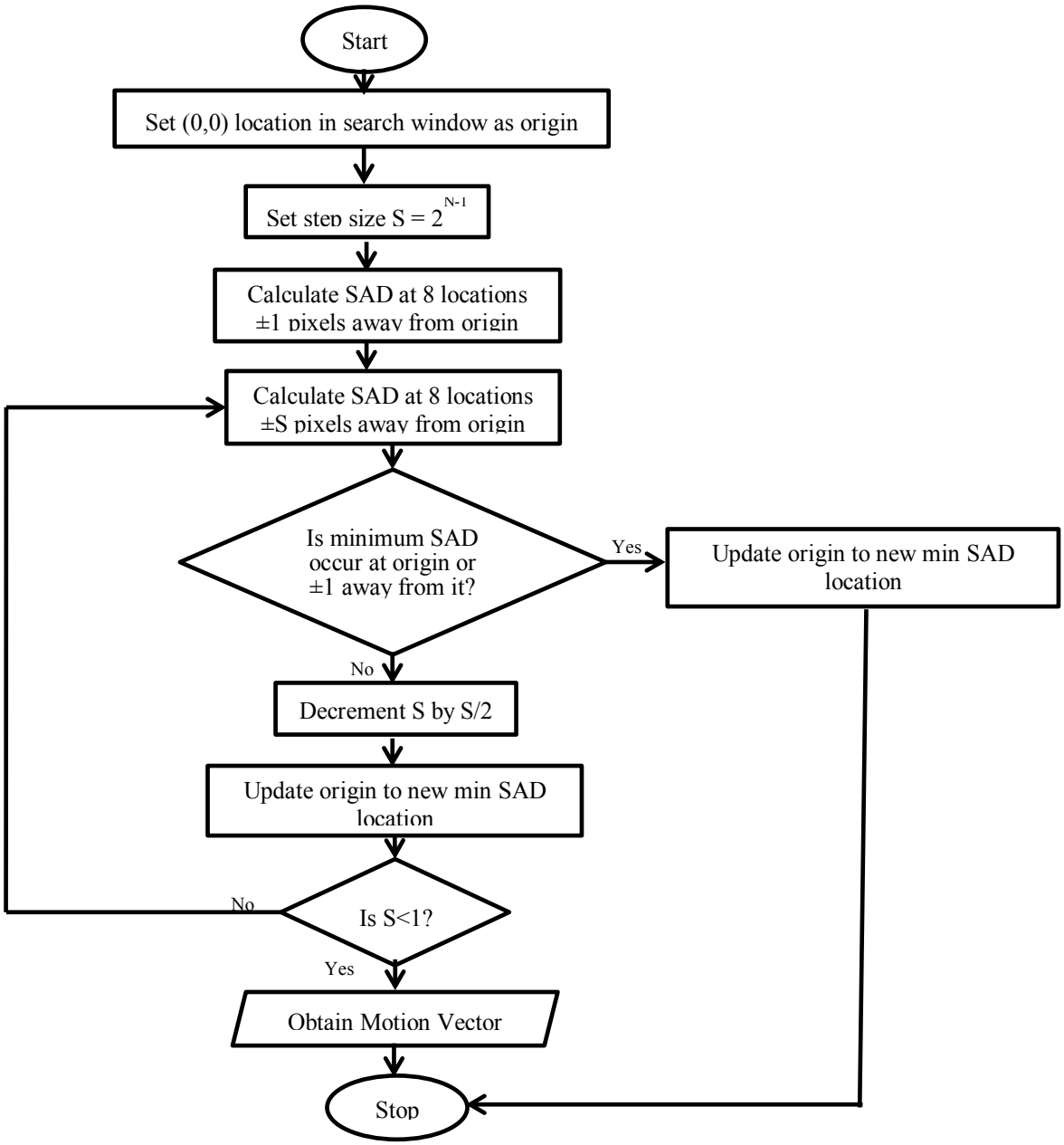

Fig.4. 5: Flow Chart of New Three Step Search Algorithm`

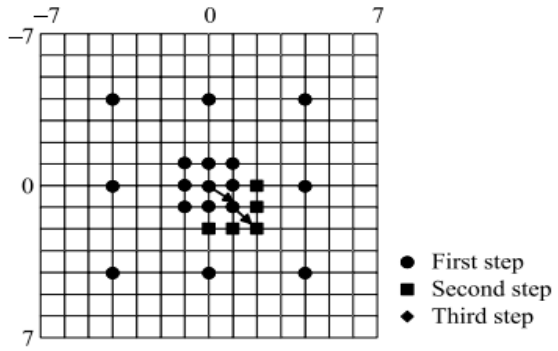

Fig.4. 6: Illustration of NTSS Algorithm

C.C. = 17 for early termination

= 9+8×($\log_2$(N+1)) otherwise.

4.5 Four Step Search (FSS) Algorithm

Since NTSS adds additional computational complexity FSS is invented to cope up with the problem. FSS [10] uses the property of center-biased motion vector distribution. It attains similar quality as NTSS reducing the computational complexity. The procedure is represented in the flow chart fig.4.7 and described below:

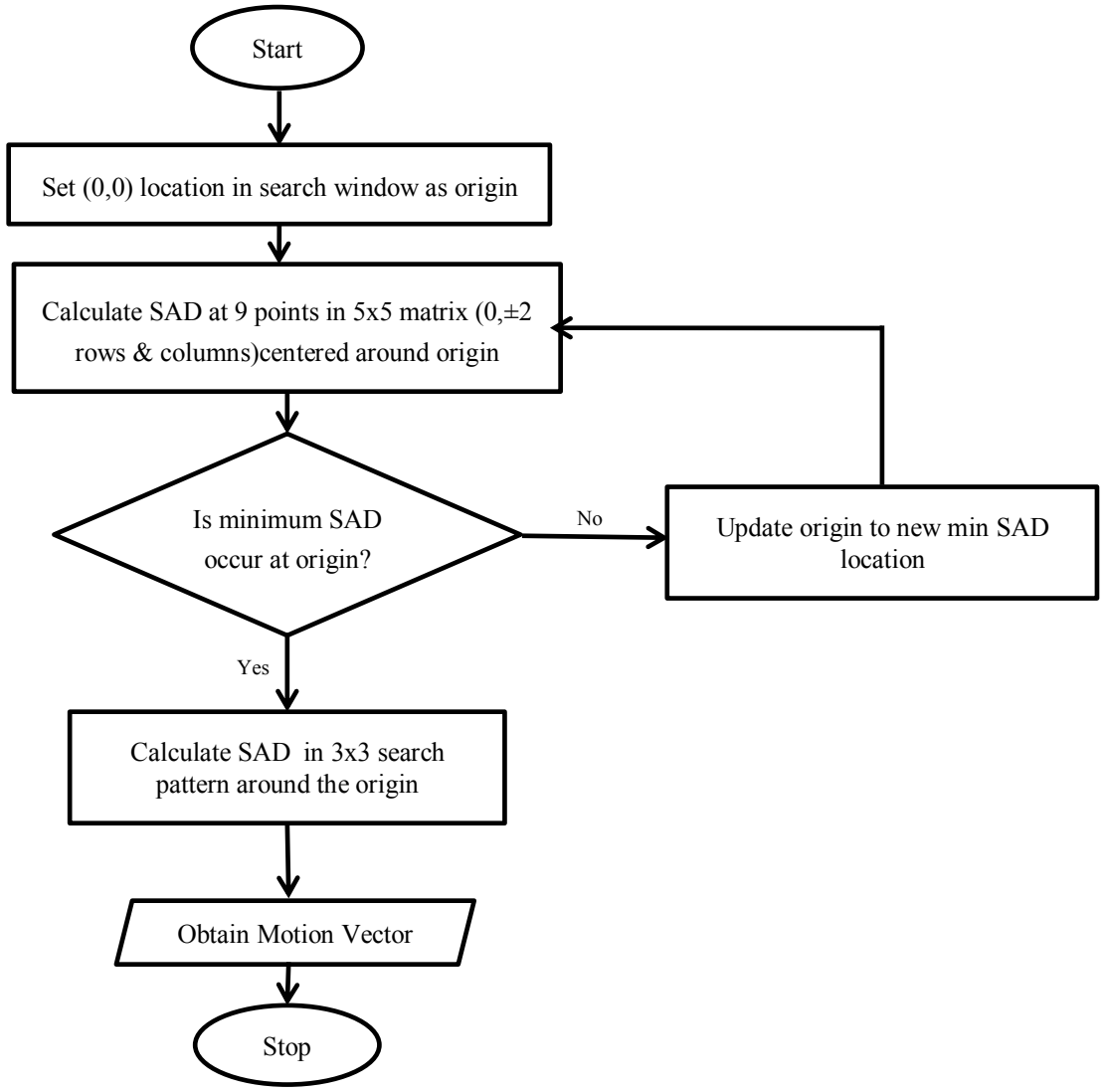

Fig.4. 7: Flow Chart of Four Step Search Algorithm

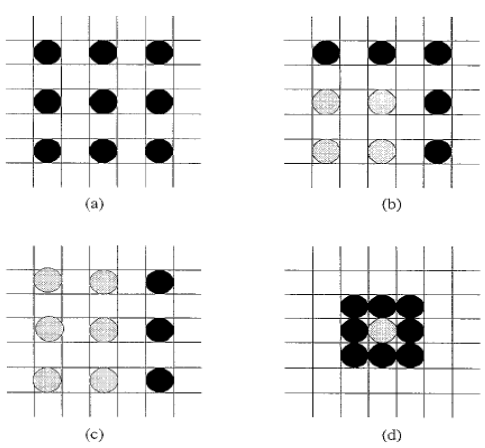

Fig.4. 8: FSS search patterns

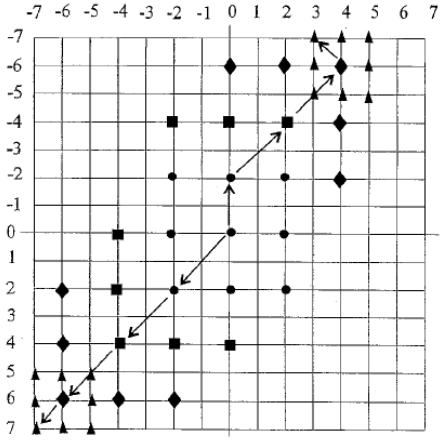

Fig.4. 9: FSS in two different paths

Step 1: Minimum SAD point is calculated from the nine search points in 5X5 search window around the center as shown in fig.4.8 (a). If it is found at the center proceed to step 3 otherwise update the center to new minimum SAD position and continue.

Step 2: (a) If the minimum point is at the corner of the previous search pattern, five additional points are added to new search pattern as shown in fig.4.8 (b).

(b) If it is found at the face center three additional points are added as shown in fig.4.8 (c). This step is repeated till the minimum SAD point occurs at the center.

Step 3: The search window is reduced to 3X3 pattern as shown in fig.4.8 (d). The nine search points are checked for minimum SAD point and the motion vector is obtained.

Fig.4.9 depicts two different search paths of FSS.

$$C.C. = 9+5\times n_1+3\times n_2+8$$

Where n_1 = no. of iterations the minimum SAD point occur at corners of 5*5 search pattern

. n_2 = no. of iterations the minimum SAD point occur at face centers of 5*5 search pattern.

In worst case the no. of search points of TSS, NTSS and FSS are 25, 33 and 27 respectively. The only disadvantage is that it is susceptible to local minimum.

4.6 Diamond Search (DS) Algorithm

Diamond Search [11] has a compact structure suitable for exploiting the center-biased characteristic of motion vector distribution- basic property of real time videos. The flow chartof the Diamond Search is shown in fig.4.10. It is described as follows:

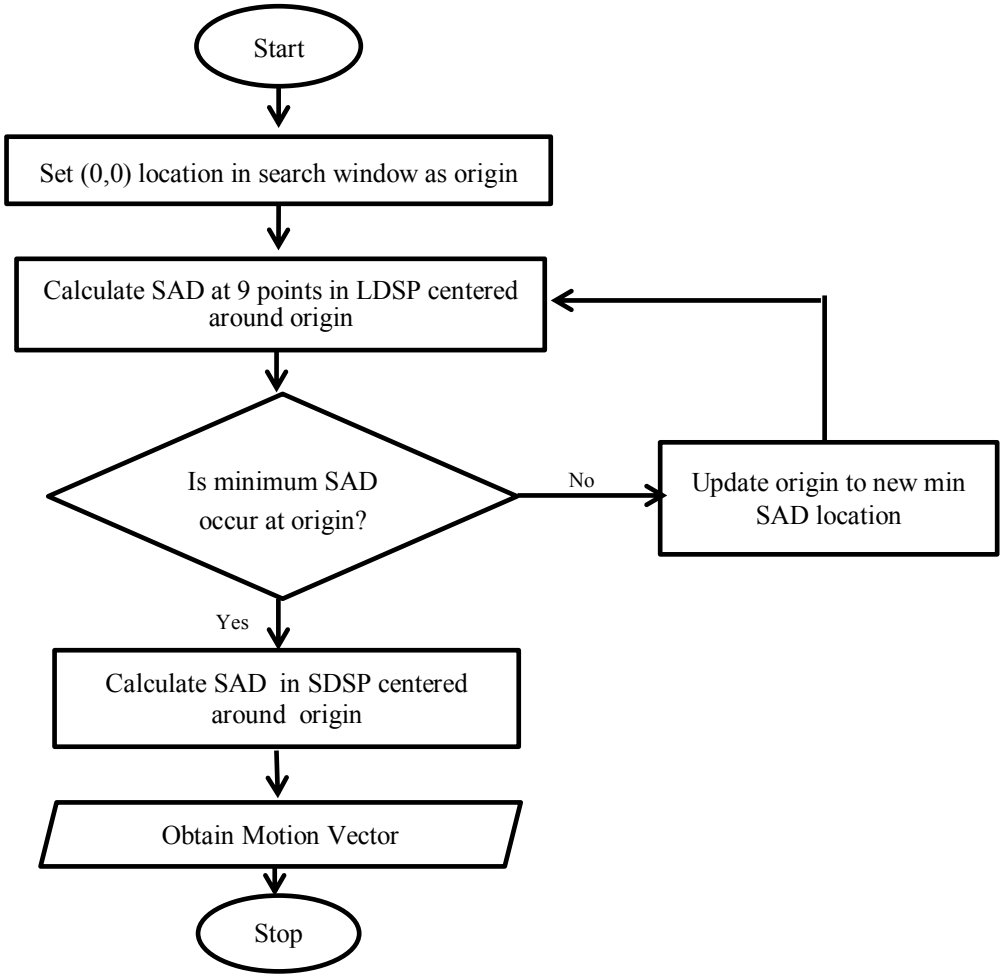

Fig.4. 10: Flow chart of Diamond Search

Step 1:The original Large Diamond Search Pattern (LDSP) is placed at the center of the search window as shown in fig.4.11 (a). The SAD for the nine points are calculated and minimum SAD point is determined. If it is found at the origin proceed to step 3, otherwise update the origin to the new minimum position and continue.

Step2: (a) If the minimum point is at the vertex of the previous search pattern, five additional points are added to new search pattern as shown in fig.4.11 (b).

(b) If it is found at the face center three additional points are added as shown in fig.4.11 (c).

This step is repeated till the minimum SAD point occurs at the origin.

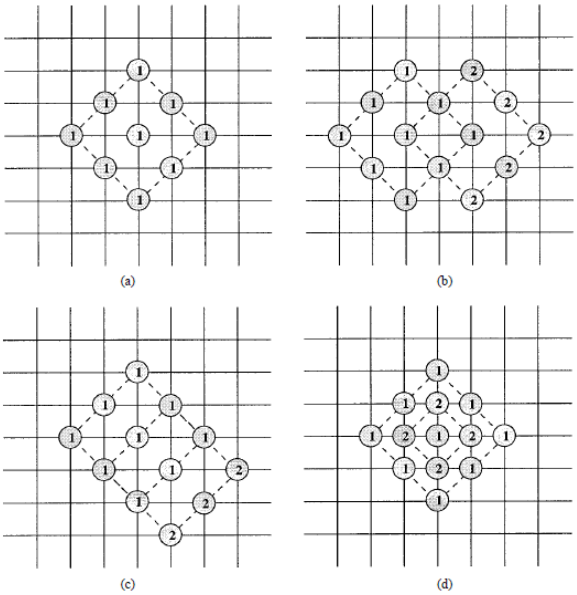

Fig.4. 11: Diamond Search Patterns

(a) LDSP (b) Vertex Search (c) Face Search (d) 1-LDSP, 2-SDSP

Step 3: The inner diamond search is performed using Small Diamond Search Pattern (SDSP) as shown in fig.4.11 (d). The four search points are checked for minimum SAD point and the motion vector is obtained.

DS attempts to reach out as far as possible in each step to search for larger motion blocks. Such a strategy is critical to reduce susceptibility to being trapped in local optimum.

C.C. = $9+5\times n_1+3\times n_2+4$

Where n_1 = no. of iterations the minimum SAD point occur at vertices of LDSP.

. n_2 = no. of iterations the minimum SAD point occur at face centres of LDSP.

4.7 Cross Diamond Search (CDS) Algorithm

DS performs computations at 13 points to decide in case it is a stationary block. To reduce the computational complexity CDS [12] is introduced. It employs Cross Shaped Pattern consisting of nine search points as shown in fig.4.12. The procedure is depicted in the fig.4.13.

Step 1: SAD s calculated for nine points in the CSP. If the minimum SAD point occurs at the origin the search is terminated (first-step termination) otherwise proceed to next step.

Step 2:(a).Two additional points closest to the current minimum point are checked i.e. two of four points located at (±1, ±1). This is called Half-Diamond Search pattern shown in fig.4.12.

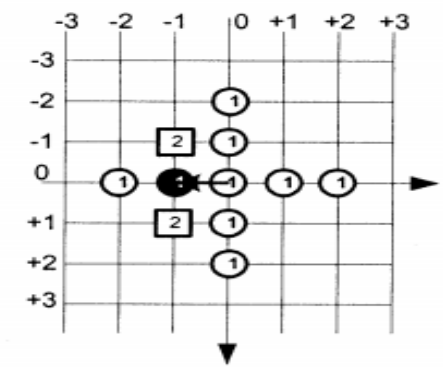

Fig.4. 12: 1-CSP 2-Half Diamond Pattern

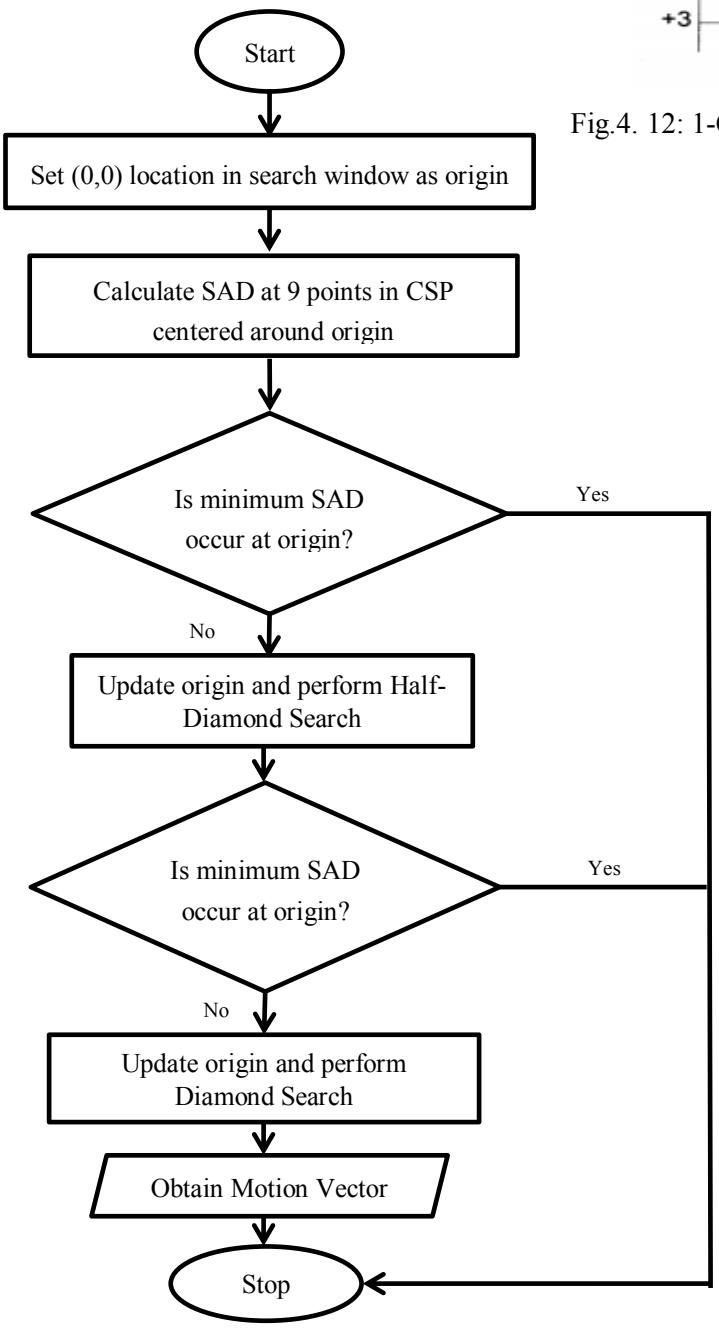

Fig.4. 13: Flow Chart of Cross- Diamond Search

(b). If the new minimum point coincides with the old one the search is stopped (second step termination). Otherwise continue.

Step 3: Update the origin to new minimum SAD position and perform Diamond Search.

C.C. = 9 for 1st step termination

=11 for 2nd step termination

$=11+4+5\times n_1+3\times n_2+4$ otherwise

where n_1 and n_2 are same as in DS.

4.8 Kite Cross Diamond Search (KCDS) Algorithm

KCDS is an improvement over CDS in order to fit small center-biased characteristics of the videos. It uses small cross shaped pattern (SCSP) in the first step which further reduces the no. of search points required in the decision making of stationary blocks [13]. It also uses asymmetrical kite patterns shown in fig.4.14 boosting the speed of the ME algorithm. The process is depicted in the fig.4.15.

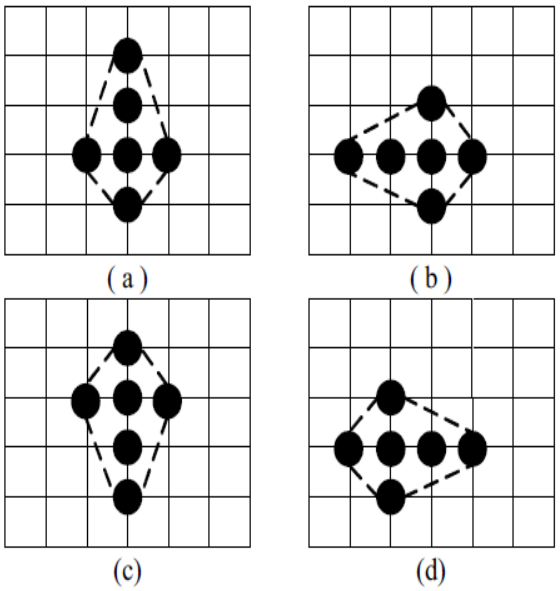

Fig.4. 14: Horizontal symmetry (a) Up-kite (c) Down-kite Vertical symmetry (b) Left-Kite (d) Right-Kite

Algorithm:-

Step 1 (SDSP): The minimum SAD point is calculated from the 5 search points of SDSP pattern centered at the updated location in the search window. If it is found at center, search is stopped otherwise the center is updated to new minimum SAD point.

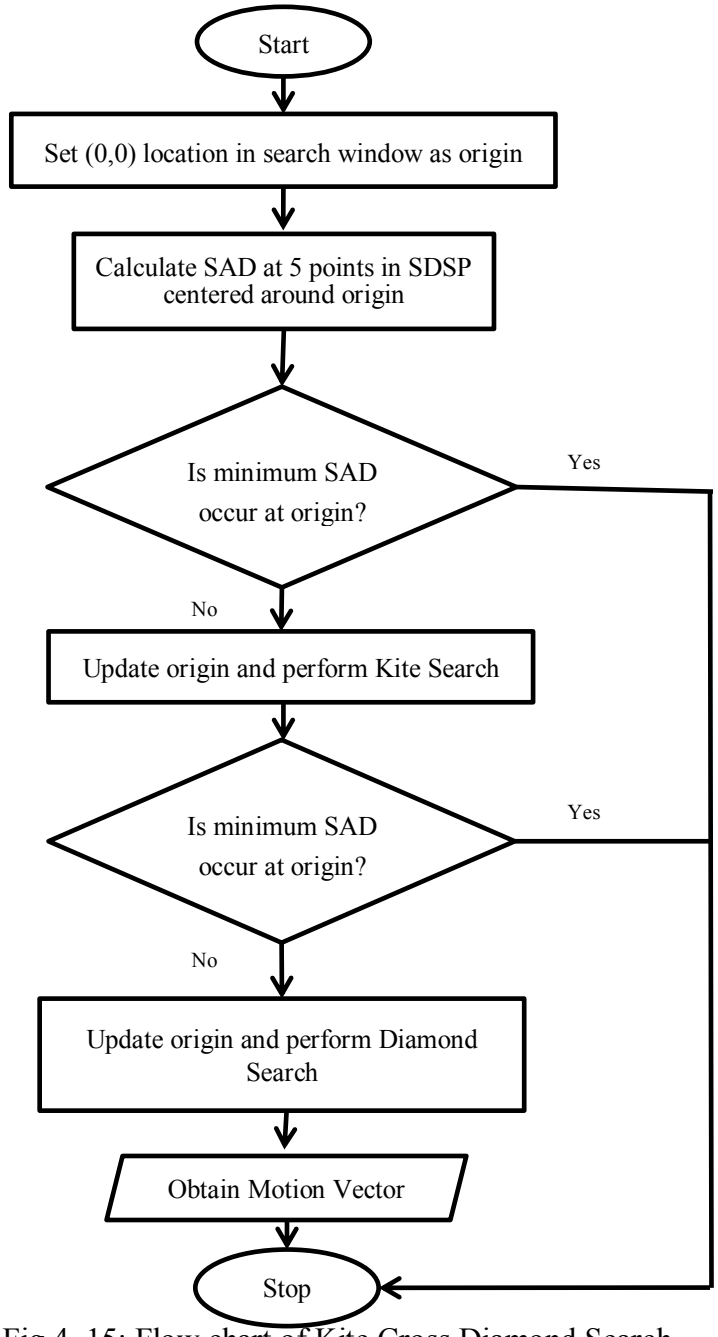

Fig.4. 15: Flow chart of Kite Cross Diamond Search

Step 2 (KSP): Based the minimum point in step 2, the corresponding kite pattern in fig.3.8 is selected. If the minimum point in previous step occurs at upper vertex of SDSP, the new KSP is an up-kite. If the minimum SAD occurs at the center of the kite, then search stops otherwise update center and proceed further.

Step 3 (LDSP): A new LDSP is formed at the updated center and SAD is calculated. If the minimum SAD point occurs at the center, proceed to step 4 else repeat step 3.

Step 4 (Inner Diamond Search): SDSP is formed around the minimum position obtained for inner diamond search. The minimum SAD point is calculated and identified as the required motion vector.

C.C. = 5 for 1st step termination

=9 for 2nd step termination

=9+k+5*n_1+3*n_2+4 otherwise

Where n_1 and n_2 are same as in DS and k denotes the no. of new search points added during selection of first diamond.

4.9 Hexagonal Search (HEXS) Algorithm

Ideally, a circle-shaped search pattern with a uniform distribution of minimum number of search points is desirable to achieve the fastest search speed. Diamond pattern which is just a 90^0 rotation to the square pattern does not meet our requirements. The Hexagon based search pattern (HEXSP) [14] may be the approximate solution for this. It is illustrated in the fig. 4.16 (1).

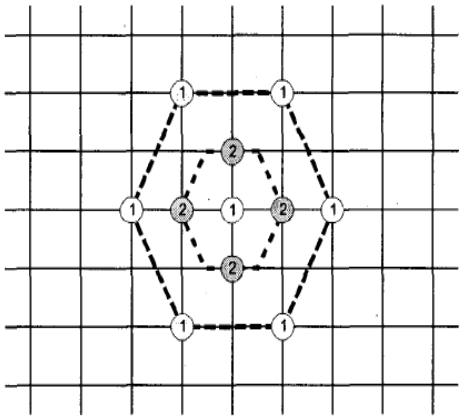

Fig.4. 16: 1-Large Hexagonal Pattern 2-Inner Hexagonal Pattern

The procedure of Hexagonal Search is depicted in the fig.4.17 and described as follows:

Step 1: The large hexagon is centered at (0,0) and checked at seven search points. If the minimum SAD is found at the center proceed to step 3 else continue.

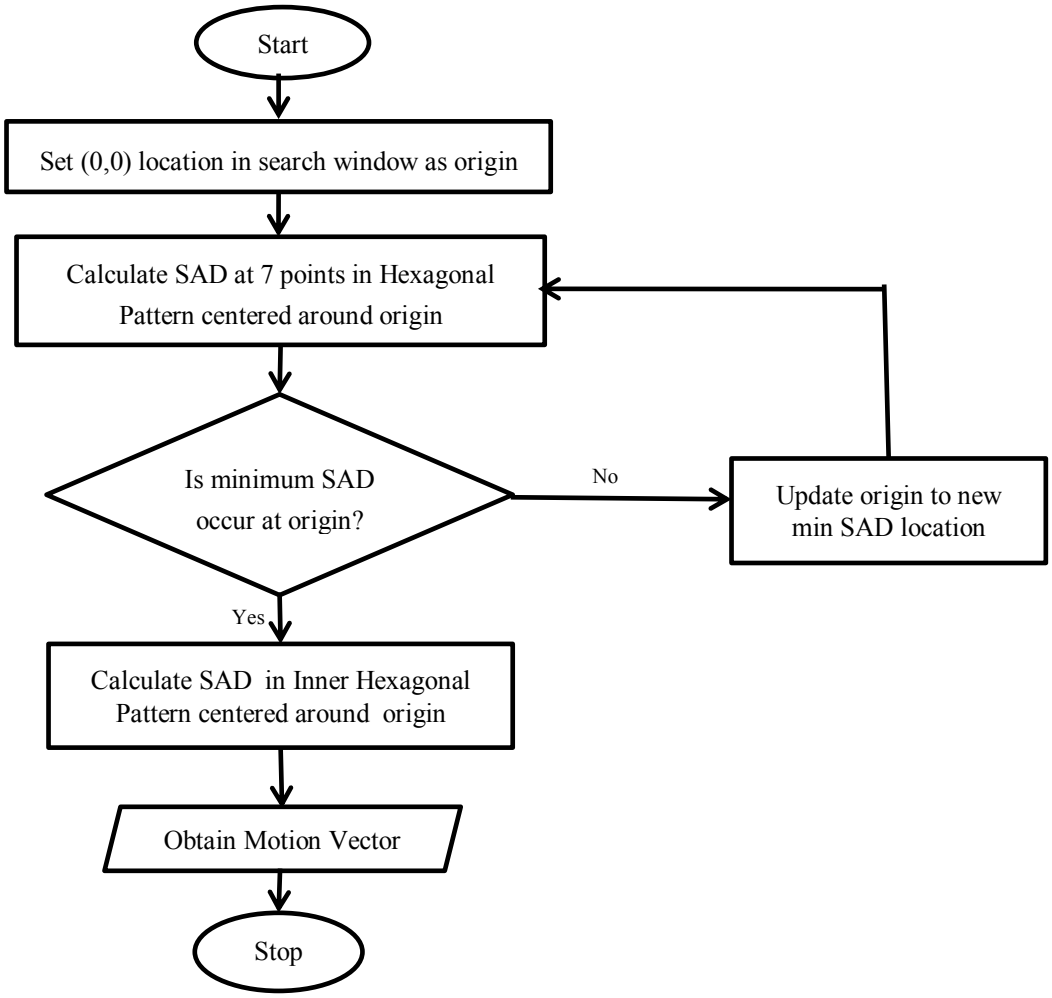

Fig.4. 17: Flow Chart of Hexagonal Search

Step 2: Update the center to a new minimum SAD point and repeat the step 1 till the minimum is found at the center. In this step additional three search points are added for each iteration.

Step 3: Switch the search pattern to small hexagon shown in fig.4.16 (2) and calculate the SAD at four points. The minimum SAD position is the final solution of our motion vector.

C.C. = 7+3*n+4

Where n=no. of iterations the hexagon pattern is repeated.

4.10 Enhanced Hexagonal Search (ENHEXS) Algorithm

ENHEXS Algorithm [15] is an improvement over HEXS algorithm employing motion vector prediction (MVP) and efficient inner hexagonal search. It utilizes the property of spatial correlation of motion vectors in practical videos as shown in fig.4.18. In this algorithm the motion vector of current block (CB) is predicted using the motion vectors of

neighbouring blocks i.e. upper block (UB) and left block (LB) in the first step. It gives a good initial point for HEXS search.

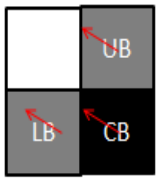

Fig.4. 18: Spatial Correlation

Further refinement of motion vector is done using hexagonal search. In HEXS search during inner hexagonal search four inner hexagon points are ignored. ENHEXS include this constraint along with reducing the no. of inner points to be searched. The flow chart of ENHEXS is depicted in fig.4.19.

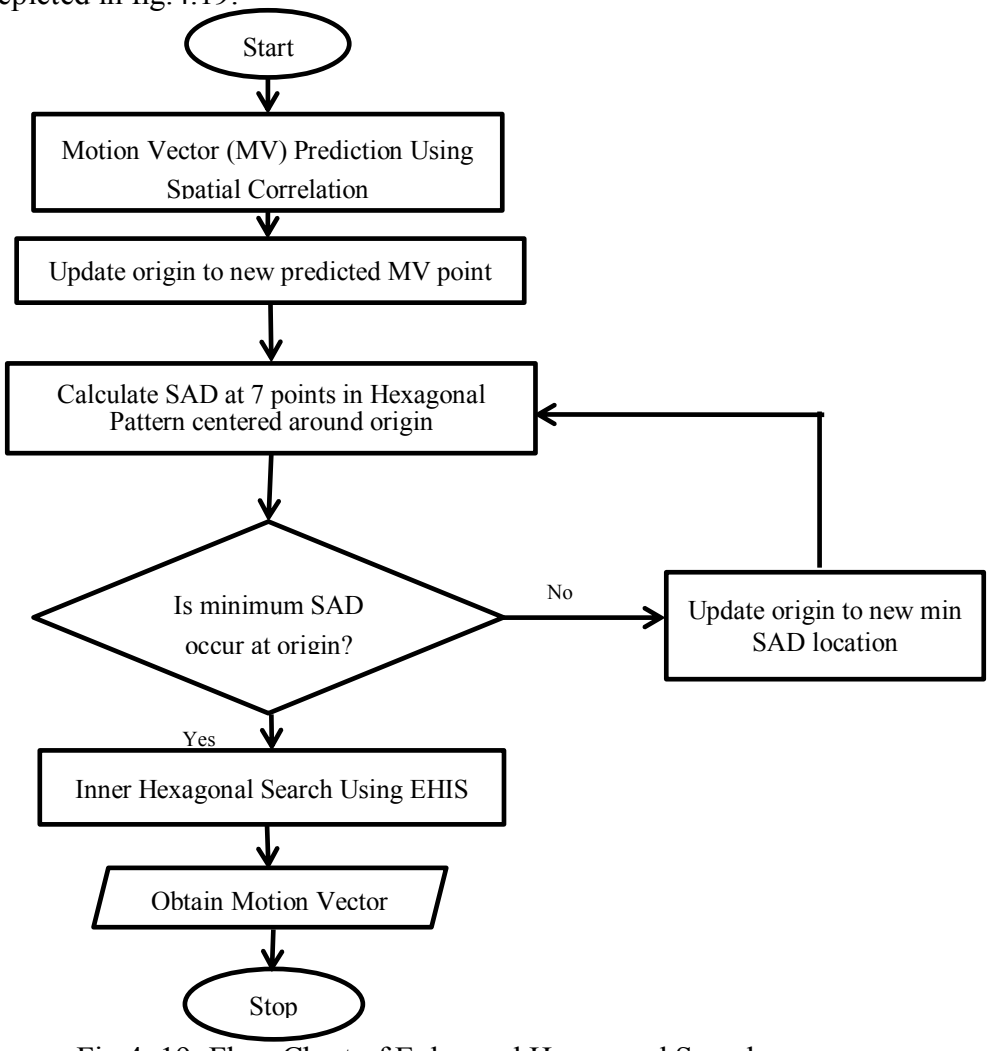

Fig.4. 19: Flow Chart of Enhanced Hexagonal Search

Algorithm:-

Step 1: Motion Vector is predicted using the motion vectors of left block, upper block and (0, 0) motion vector. SAD is calculated all the three points. The minimum SAD point is determined and the corresponding point is updated as the initial point for the search.

Step 2: Large hexagon is centered at the position of the predicted motion vector and the SAD is calculated around seven points till the minimum SAD point is obtained at the center. Then proceed to step 3.

Step 3: The hexagon is divided into six groups as shown in fig.4.20. If one of group1, group 2, group 4 or group 5 has the second minimum SAD point then only two inner points close to that group are searched. It is illustrated in fig.4.20 (a). If group 3 or group 6 wins only one point is searched as illustrated in fig.4.20 (b).

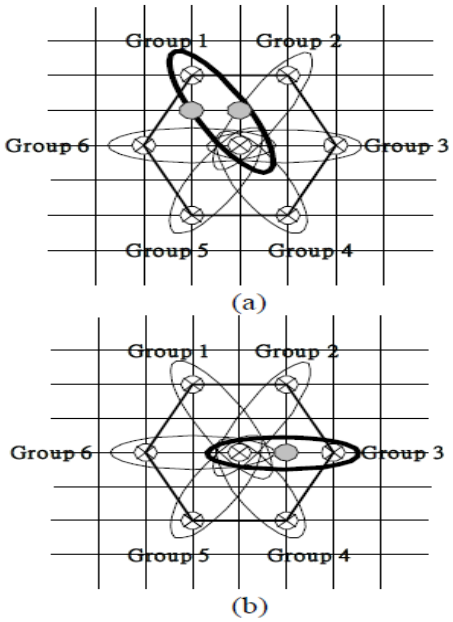

Fig.4. 20: Efficient Hexagonal Inner Search

C.C. = 3+6+3*n+1=10+3*n if group 3 or 6 wins

= 3+6+3*n+2 =11+3*n otherwise

where n=no. of iterations the hexagon pattern is repeated.

CHAPTER 5

PROPOSED MOTION ESTIMATION ALGORITHMS

5.1 Motion Vector Prediction

Correlation is a measure which defines how one quantity changes in relation to the other. In real time videos all the frames are highly correlated to each other. Generally the objects in each frame spread in more than one block. Hence when the object moves the motion of all the blocks where the object is present move in the similar direction. This tendency of a motion vector of a block related to its neighbouring blocks is known as spatial correlation which is depicted in fig.4.18. In temporal domain MV is also correlated to its corresponding collocated block in previous frame. This property is known as temporal correlation depicted in fig.5.1.

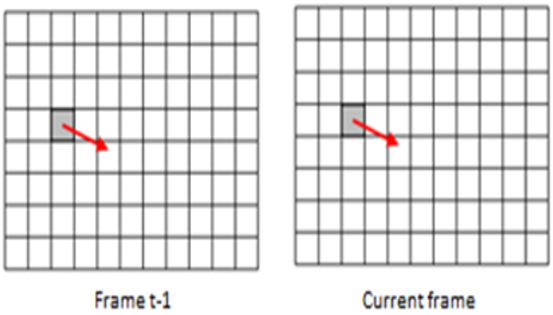

Fig.5. 1: Temporal Correlation

The concept of predicting the motion vector of the current block using the motion vectors of the neighbouring blocks in the current frame and collocated block of the previous frame is known as "Motion Vector Prediction (MVP)" [3]. This highly reduces the number of search points to be checked which implies the reduction on computational complexity of the algorithm.

5.1.1 Modified ENHEX Search (MENHEXS) Algorithm

In ENHEXS algorithm only spatial correlation is considered in motion vector prediction. In our algorithm, temporal correlation property [9] shown in fig.5.1 is also exploited. Hence motion vector is predicted based on the using the motion vectors of left

block, upper block and (0, 0) motion vectors of the current frame and the motion vector of the collocated block in the previous frame. The procedure is depicted in the flow chart fig. 5.2 and described as follows:

Step 1: The position of minimum SAD value is calculated from the five search points in the prediction set and the output is considered as the predicted motion vector and the origin of the pattern is updated to that position.

Step 2: Then Hexagonal search is adapted for further accuracy of motion vector till the minimum SAD point occur at origin of the pattern.

Step 3: Now the inner hexagon is searched efficiently using the EHIS pattern shown in fig. 4.20.

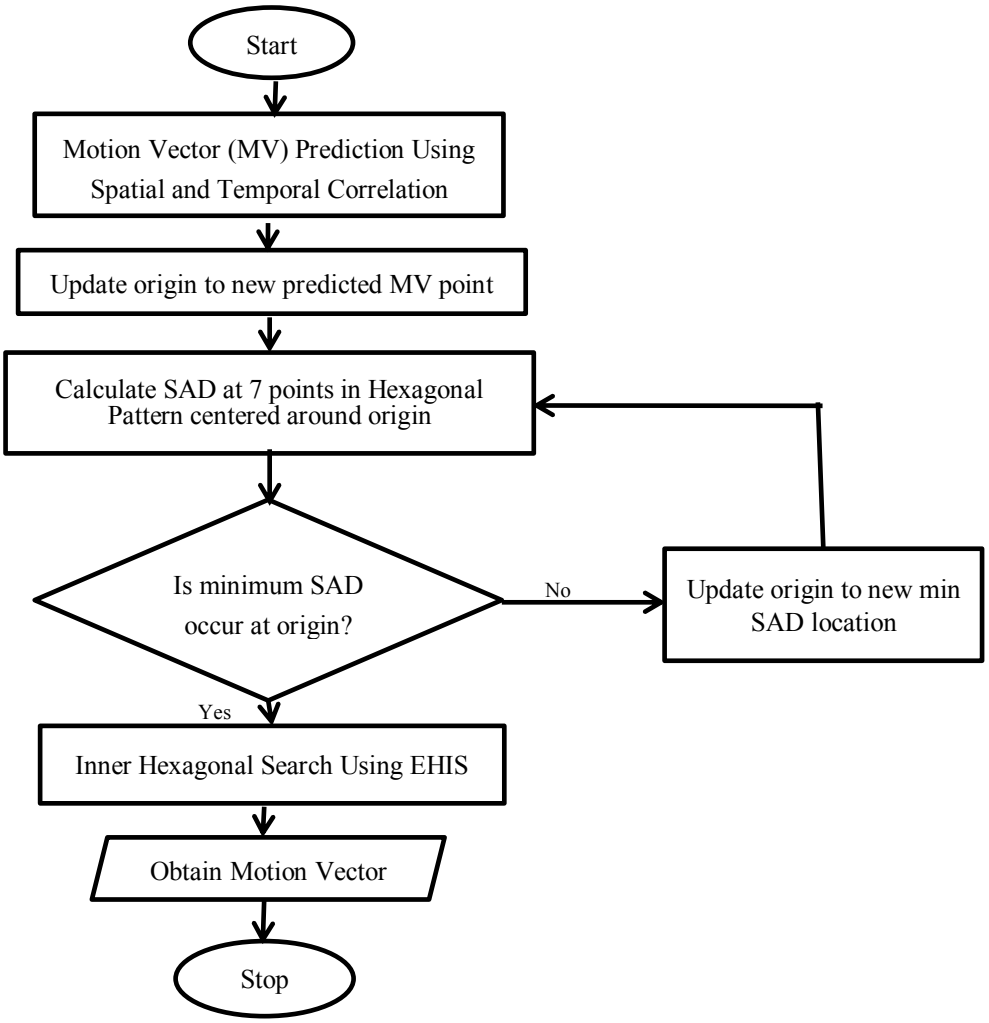

Fig.5. 2: Flow Chart of Modified Enhanced Hexagonal Search

5.1.2 Enhanced KCDS (ENKCDS) Algorithm

Analysing the above algorithms mentioned in chapter 4 it is found that the KCDS algorithm performs fast with very low computational complexity and acceptable PSNR.

Hence motion vector prediction part based on spatial correlation is added to this algorithm for better results. The flow chart in fig. 5.3 portrays the procedure of Enhanced KCDS algorithm.

In this algorithm, motion vector prediction is done using spatial correlation. Then kite cross diamond search is adapted for précising the attained motion vector.

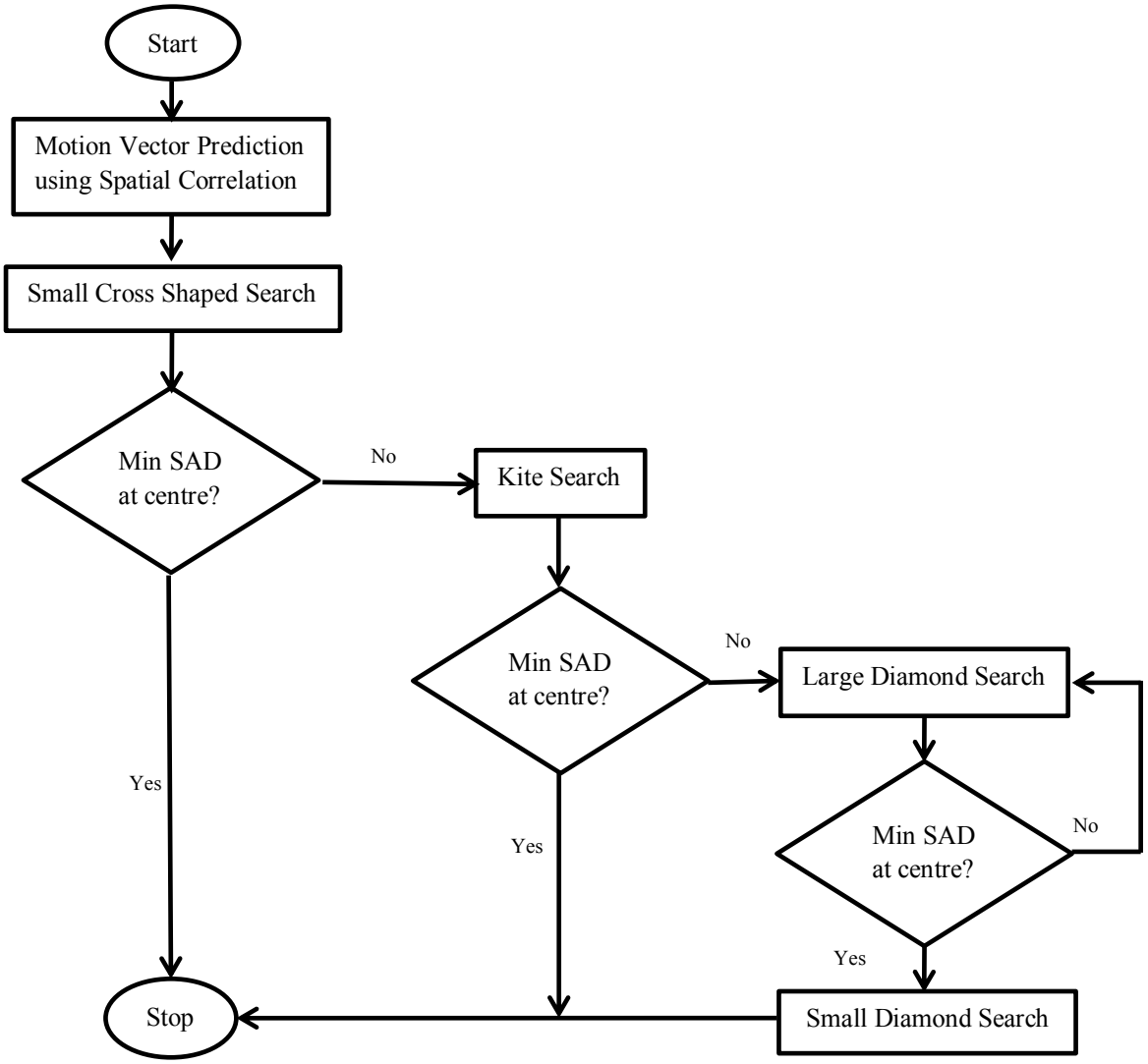

Fig.5. 3: Flow Chart of Enhanced Kite Cross Diamond Search Algorithm

5.1.3 Modified ENKCDS (MENKCDS) Algorithm

MENKCDS Algorithm is an improvement over ENKCDS search. In this algorithm, motion vector prediction includes temporal correlation along with the spatial correlation. Fig.5.4 depicts the procedure of this algorithm.

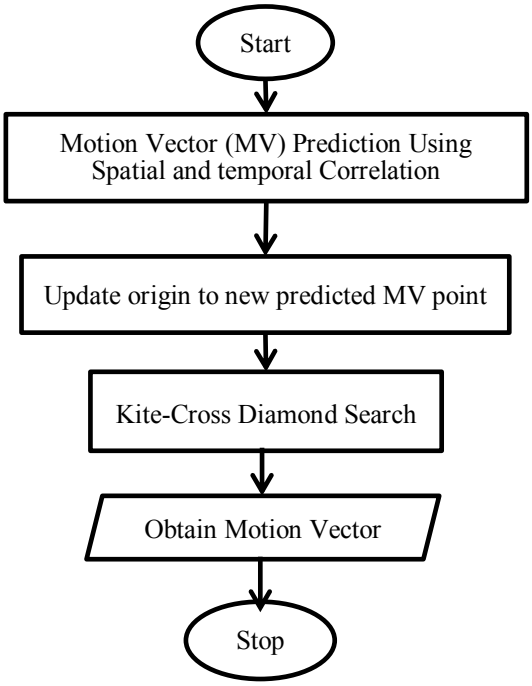

Fig.5. 4: Flow Chart of Modified Enhanced KCDS Algorithm

5.1.4 Hybrid Hexagonal Kite Cross Diamond Search (HYBHKS) Algorithm

HYBHKS Algorithm uses motion vector prediction in the initial step for initial classification of the motion type of the block. Then minimum SAD value is compared with threshold T_1 to classify it as quasi-stationary. If not Small Diamond Search pattern is used and searched at 5 points. If the minimum SAD is at centre, then the search is terminated. Otherwise based on the minimum SAD value, the motion is classified as small motion or large motion and different search patterns like KCDS and HEXS are performed respectively. The flow chart of HYBHKS is shown in fig.5.5.

The algorithm is described as follows:

Step 1(MV Prediction):- SAD is calculated at four search points obtained from the motion vectors of upper block (UB), left block (LB), zero motion vector of current frame and collocated block in the previous frame. The minimum SAD position is calculated and the

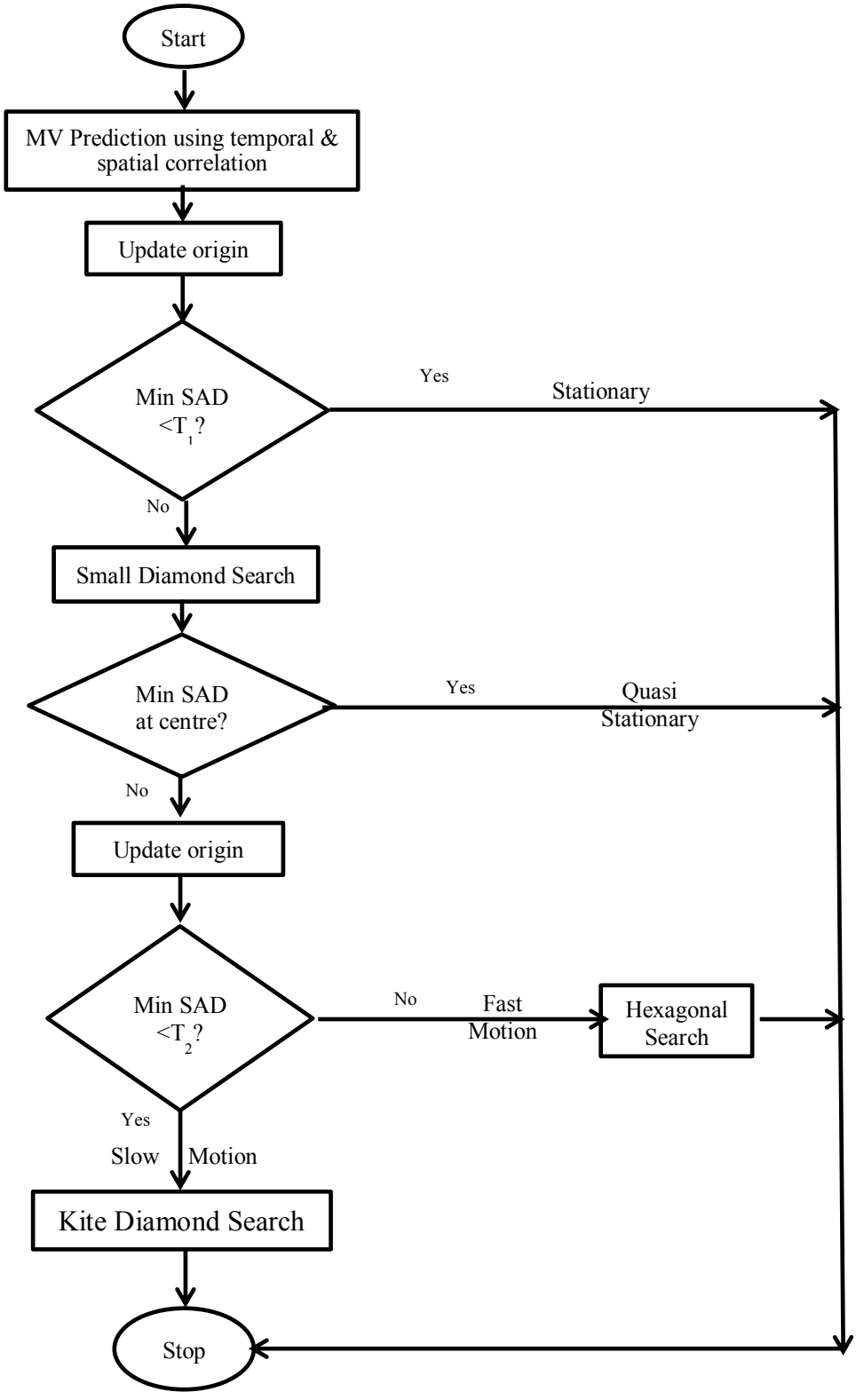

Fig.5. 5: Flow Chart of HYBHKS Algorithm

origin is updated to that position. If the minimum SAD value is less than T_1, the block is classified as quasi-stationary and the search is terminated. Otherwise, go to step 2.

Step 2 (SDS):- The minimum SAD point is calculated from the 5 search points of SDSP pattern centered at the updated location in the search window. If it is found at center, search is stopped otherwise the center is updated to new minimum SAD point and proceed to next step.

Step 3 (Motion Type Judgment):- The minimum SAD value in step 2 is compared with the threshold T_2. If it is less than T_2 the motion of the block is judged as slow motion and the search proceeds to step 4. Otherwise it is classified as large motion and step 5 is performed.

Step 4 (KDS) [4]:-

(a). Kite Search:- Based on the direction of the minimum SAD position in SDSP, the corresponding directional kite is selected from the KSP shown in fig.4.14 and searched at 6 positions. If the minimum SAD is obtained at the centre of the kite, the search is terminated else proceed to step 4(b) after updating the center.

(b). Diamond Search: - SAD is calculated at nine points of LDSP shown in fig.4.11. The minimum SAD is calculated and the center is updated to the minimum position. This is repeated till the minimum position is found at the centre. Then SDSP is performed for diamond inner search.

Step 5 (HEXS) [5]:-

(a). Hexagonal Search:- SAD is calculated at the seven points of HSP shown in fig.4.16. The centre is updated to the minimum SAD position. This step is repeated till the minimum position is found at the center. Then proceed to step 5(b).

(b). EHIS:- The hexagon is divided into six groups as shown in fig.4.20. If one of group1, group 2, group 4 or group 5 has the second minimum SAD point then only two inner points close to that group are searched. It is illustrated in fig.4.20 (a). If group 3 or group 6 wins only one point is searched as illustrated in fig.4.20 (b).

5.2 Results and Discussion

5.2.1 Comparison of various ME Algorithms

Various experiments are conducted to verify the performance of the proposed method. CIF (352×288) video sequences akiyo, foreman, ice and mobile are taken as test vectors. All experiments are performed using MATLAB version 7.0.10.499 (R2010a) on an Intel® core(TM) i5-2400 CPU @ 3.10GHz.

The performance of the various ME algorithms is measured by inspecting the performance metrics discussed in chapter 3. The optimum ME algorithm for the specified

application is decided according to the trade-off between the quality along with complexity and new size of the received video at the decoder. PSNR and SSIM of Y component are used in assessing the quality of video. Search points are used in the assessment of the computational complexity of the algorithms whereas Encoded bit rate and Compression Ratio are presented to decide the trade-off. Tables 5.1, 5.2 and 5.3 depicts the results of the test videos foreman, ice and akiyo respectively. The Quantization Parameter Q is set to 24. The thresholds T_1 and T_2 in HYBHKS algorithm is set to 300 and 600 respectively.

Table 5. 1 : Average PSNR, SSIM, Encoded Bit Rate (kbps), Compression Ratio, Average number of search points per macro block and Output File Size comparison for the test video Foreman_cif @25fps video

ME Algorithm	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio	Search Points	O/P Encoded File Size (bits)
INTRA	36.89	0.8893	1679.47	18.1086	NIL	20153632
FS	37.75	0.9035	1523.21	19.9663	196.628788	18278480
LS	37.69	0.9025	1546.07	19.67105	17.3187	18552816
TSS	37.66	0.9017	1558.05	19.5198	19.007963	18696568
NTSS	37.66	0.9018	1613.76	18.84595	19.32558	19365096
FSS	37.67	0.9019	1582.85	19.21395	16.734461	18994216
DS	37.71	0.9028	1536.54	19.79305	19.897214	18438464
CDS	37.73	0.9032	1547.08	19.65815	19.606406	18564984
KCDS	37.74	0.9034	1533.34	19.8344	16.056465	18400032
HEXS	37.68	0.9023	1569.23	19.38065	14.05431	18830800
ENHEXS	37.7	0.9026	1550.07	19.62025	10.894015	18600864
MENHEXS	37.7	0.9027	1541.21	19.73305	10.979167	18494520
ENKCDS	37.74	0.9035	1519.54	20.0145	12.523813	18234456
MENKCDS	**37.74**	**0.9035**	**1514.35**	**20.08305**	12.173173	18172224
HYBHKS	37.73	0.9032	1519.01	20.0215	**10.085976**	18228072

It can be seen that the quality of video is good in the proposed algorithms (MENKCDS and HYBHKS) with relatively low computational complexity maintaining high compression ratios.

Table 5. 2: Average PSNR, SSIM, Encoded Bit Rate (kbps), Compression Ratio, Average number of search points per macro block and Output File Size comparison for the test video Ice_cif @25fps video

ME Algorithm	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio (4:2:0 format)	Search Points	O/P Encoded File Size (bits)
INTRA	39.78	0.948	1098.91	27.6754	NIL	10549552
FS	40.66	0.9508	883.94	34.4058	196.628788	8485864
LS	40.6	0.9502	911.62	33.3612	18.582292	8751576
TSS	40.57	0.9497	913.19	33.30375	19.205335	8766664
NTSS	40.58	0.9498	953.99	31.87945	20.318739	9158336
FSS	40.59	0.9501	940	32.354	18.45362	9024016
DS	40.65	0.9506	886.88	34.29175	21.937489	8514088
CDS	**40.67**	**0.9509**	887.92	34.2519	19.758228	8523992
KCDS	40.66	0.9507	886.29	34.31335	17.469718	8508352
HEXS	40.6	0.9502	913.24	33.30225	14.194066	8767064
ENHEXS	40.61	0.9503	913.69	33.2857	12.956166	8771424
MENHEXS	40.61	0.9503	910.56	33.40015	13.134407	8741360
ENKCDS	40.65	0.9506	885.04	34.36305	17.013131	8496416
MENKCDS	40.65	0.9506	**884.12**	**34.39905**	16.878346	**8487528**
HYBHKS	40.62	0.9504	895.96	33.94455	**11.101852**	8601168

From table 5.2, it can be seen that the quality of video is good in the case of CDS algorithm but the complexity of this algorithm is very high whereas in ENHEXS, MENHEXS and HYBHKS algorithms the complexity is relatively very low maintaining a slight compromise in PSNR and SSIM values. Also the proposed algorithms ENKCDS and MENKCDS reach low complexity as compared to KCDS algorithm.

Table 5. 3: Average PSNR, SSIM, Encoded Bit Rate (kbps), Compression Ratio, Average number of search points per macro block and Output File Size comparison for the test video Akiyo_cif @25fps video

ME Algorithm	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio (4:2:0 format)	Search Points	O/P Encoded File Size (bits)
INTRA	40.11	0.9415	1150.76	26.42835	NIL	13809160
FS	41.28	0.949	651.79	46.6604	196.628788	7821488
LS	41.27	0.949	652.89	46.5817	14.928712	7834696
TSS	41.27	0.949	652.99	46.575	18.976448	7835824
NTSS	41.27	0.949	658.45	46.18845	21.529074	7901408
FSS	41.27	0.949	653.53	46.53605	14.135842	14.135842
DS	41.28	0.949	651.94	46.6494	13.42601	7823328
CDS	41.28	0.9491	652.41	46.61615	10.907264	7828912
KCDS	41.28	0.949	651.91	46.652	**7.963316**	7822896
HEXS	41.27	0.949	653.21	46.55925	9.42287	7838480
ENHEXS	41.27	0.949	653.37	46.54795	9.474175	7840440
MENHEXS	41.28	0.949	653.1	46.56675	10.092399	7837216
ENKCDS	41.28	0.949	651.89	**46.65315**	9.7317	7822696
MENKCDS	41.28	0.949	651.9	46.6527	10.334992	7822776
HYBHKS	41.28	0.949	651.91	46.6521	8.153519	7822880

For this video, it is observed that the quality is similar for most of the ME algorithms. Hence the main relevance is given to compression ratio and search points. Regarding the search points view KCDS perform well whereas considering the compression ratio ENKCDS is better. HYBHKS also performs well with low search points and considerably high compression ratio.

All the above videos are standard videos. ME algorithms are also compared using a real time video where two persons are playing with flowers throwing them upwards. The results are tabulated in table 7.4.

Table 5. 4: Average PSNR, SSIM, Encoded Bit Rate (kbps), Compression Ratio, Average number of search points per macro block and Output File Size comparison for a real time video @5fps video

ME Algorithm	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio (4:2:0 format)	Search Points	O/P Encoded File Size (bits)
FS	37.22	0.9073	361.57	16.8229	196.628788	2892520
KCDS	37.2	0.9071	361.26	16.8372	17.124369	2890056
HEXS	37.19	0.9067	365.99	16.6197	14.749495	2927888
ENHEXS	37.18	0.9066	365.84	16.6262	12.38971	2926728
MENHEXS	37.19	0.9066	365.02	16.6635	12.44072	2920176
ENKCDS	37.2	0.9071	359.96	**16.8979**	14.928346	2879672
MENKCDS	37.2	0.9071	359.97	16.8974	14.637689	2879760
HYBHKS	37.2	0.907	361.71	16.8163	**12.232071**	2893648

Hence from the above tables it is noticed that in most of the videos the proposed algorithms perform pretty well in all aspects. ENKCDS and MENKCDS have less number of search points when compared to KCDS algorithm. HYBHKS performance is good in almost all videos irrespective of the motion type (like fast motion in ice_cif and slow motion in foreman_cif and akiyo_cif videos).

5.2.2 Optimization of threshold values

The proposed HYBHKS algorithm is verified using various threshold values. Average PSNR, SSIM and search points are calculated and tabulated in the table 4. The set of thresholds for which the search points is minimum with similar quality parameters is chosen as the optimum values for the proposed algorithm. Fig.5.6 (a)-(d) depict the comparison of the PSNR and search points for various set of thresholds.

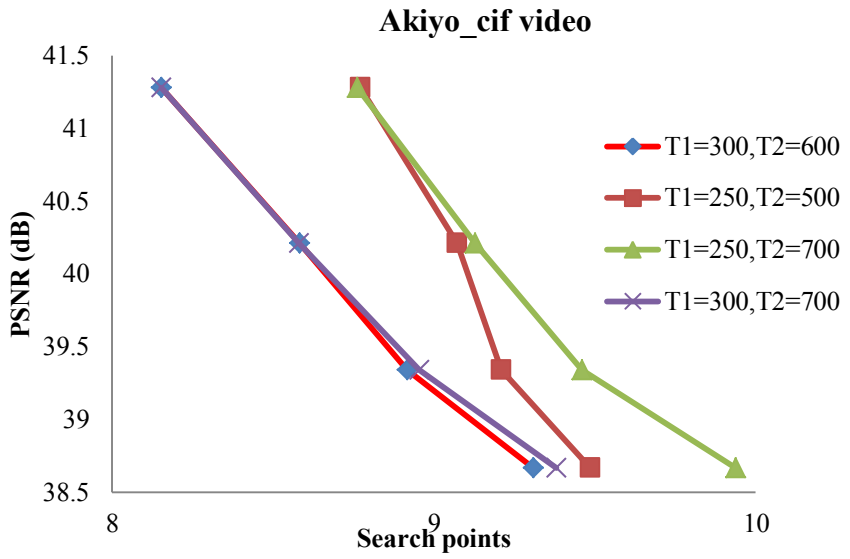

Fig.5.6 (a)

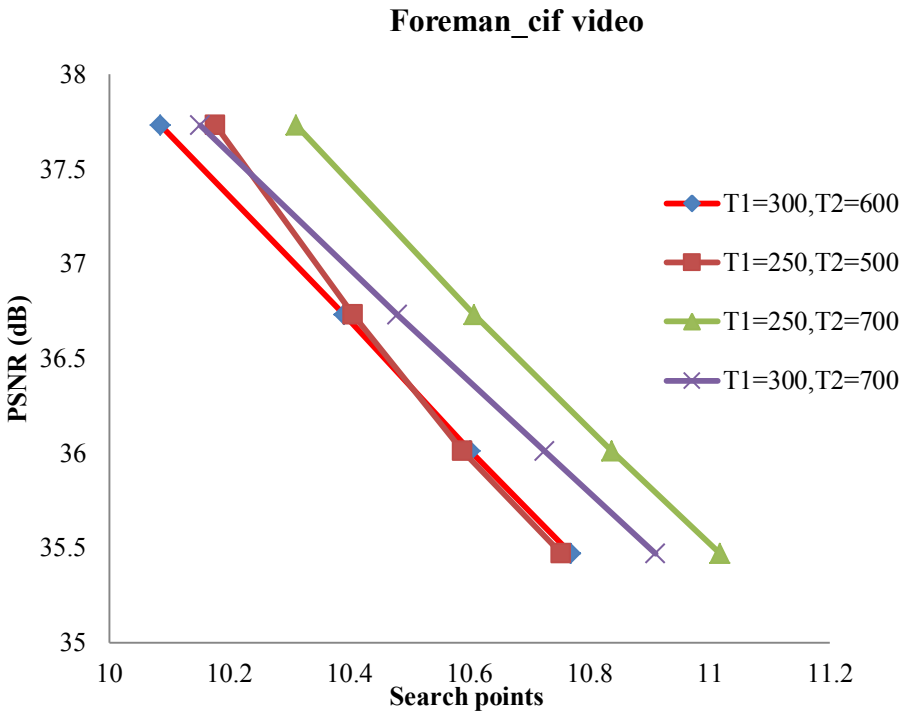

Fig. 5.6 (b)

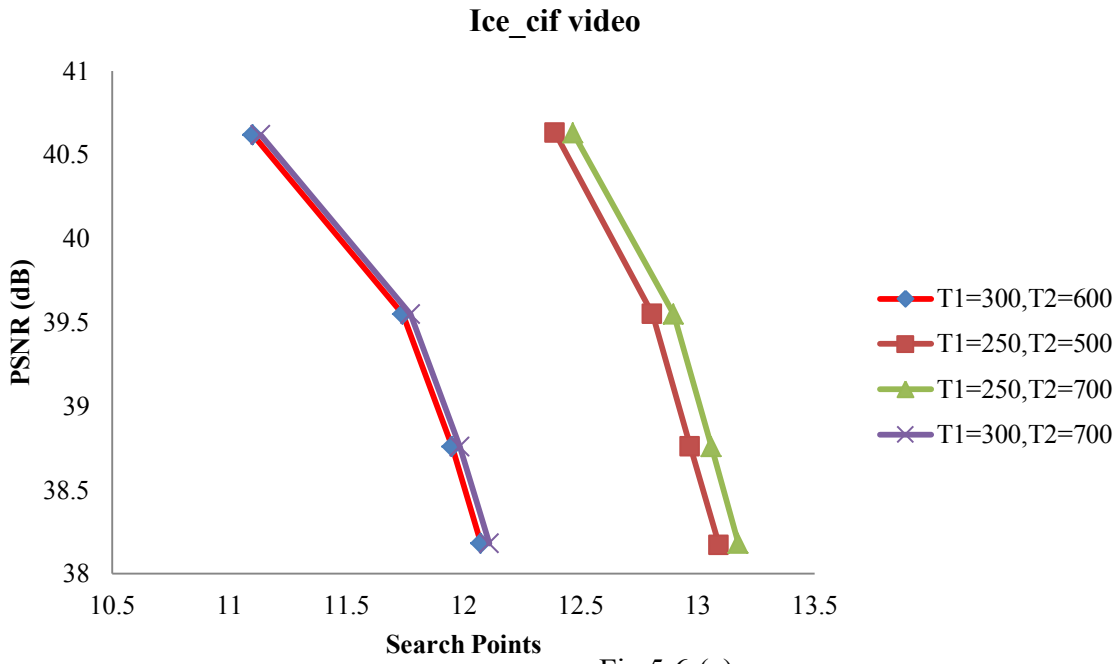

Fig.5.6 (c)

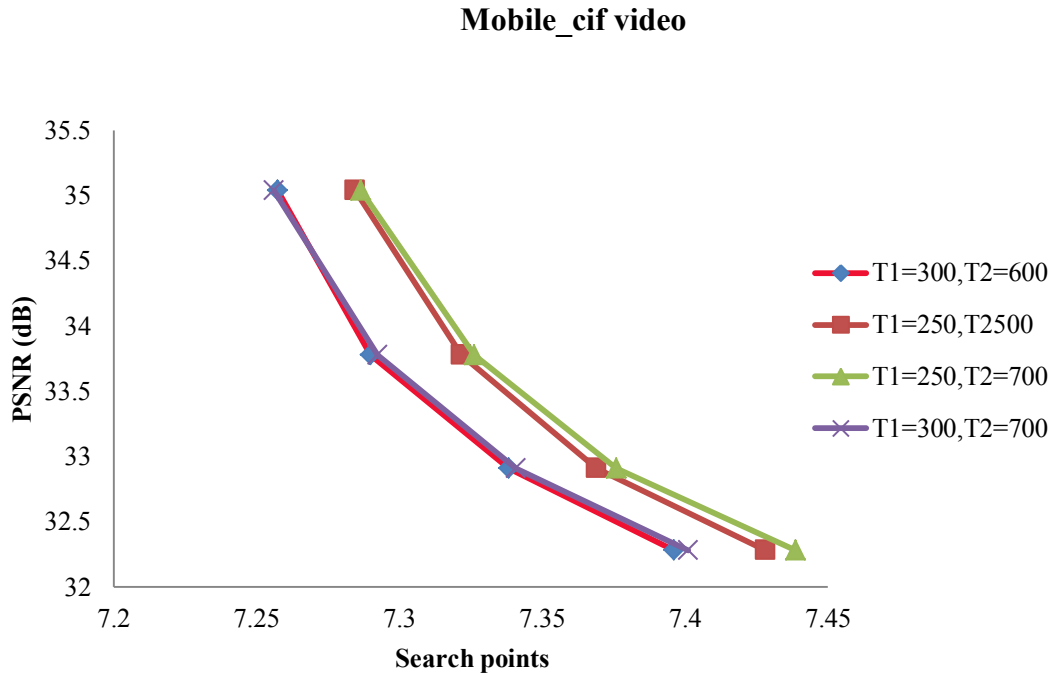

Fig.5.6 (d)

Fig.5. 6: Graph of PSNR versus Search points comparing HYBHKS algorithm for various set of threshold values

Table 5. 5: Average PSNR, SSIM and search points of HYBHKS algorithm using various threshold sets (T1=300 T2=600, T1=250 T2=500, T1=250 T2=700 AND T1=300 T2=700) for the test videos Akiyo_cif, Foreman_cif, Ice_cif and Mobile_cif.

TEST VIDEO	Q	T1=300, T2=600			T1=250, T2=500			T1=250, T2=700			T1=300, T2=700		
		PSNR (dB)	SSIM	Search Points	PSNR (dB)	SSIM	Search Points	PSNR (dB)	SSIM	Search Points	PSNR (dB)	SSIM	Search Points
AKIYO	Q=24	41.28	0.949	**8.15**	41.28	0.949	8.77	41.28	0.949	8.76	41.28	0.949	8.15
	Q=30	40.21	0.94	**8.58**	40.21	0.94	9.07	40.21	0.94	9.13	40.21	0.94	8.58
	Q=36	39.34	0.9295	**8.92**	39.34	0.9295	9.21	39.34	0.9295	9.46	39.34	0.9295	8.95
	Q=42	38.67	0.9207	**9.31**	38.67	0.9207	9.49	38.67	0.9208	9.94	38.67	0.9208	9.38
FOREMAN	Q=24	37.73	0.9032	**10.09**	37.73	0.9032	10.18	37.73	0.9032	10.31	37.73	0.9032	10.15
	Q=30	36.73	0.8846	**10.39**	36.73	0.8846	10.41	36.73	0.8846	10.61	36.73	0.8846	10.48
	Q=36	36.01	0.8676	10.60	36.01	0.8676	**10.59**	36.01	0.8676	10.84	36.01	0.8676	10.73
	Q=42	35.47	0.8527	10.76	35.47	0.8527	**10.75**	35.47	0.8527	11.02	35.47	0.8527	10.91
ICE	Q=24	40.62	0.9504	**11.10**	40.63	0.9504	12.39	40.63	0.9504	12.47	40.62	0.9504	11.13
	Q=30	39.55	0.9416	**11.74**	39.55	0.9417	12.81	39.55	0.9417	12.90	39.55	0.9416	11.77
	Q=36	38.76	0.9335	**11.95**	38.76	0.9335	12.97	38.76	0.9335	13.06	38.76	0.9335	11.98
	Q=42	38.18	0.9265	**12.08**	38.17	0.9265	13.09	38.18	0.9266	13.17	38.18	0.9265	12.11
MOBILE	Q=24	35.04	0.9314	**7.26**	35.04	0.9314	7.28	35.04	0.9314	7.29	35.04	0.9314	7.26
	Q=30	33.78	0.9115	**7.29**	33.78	0.9115	7.32	33.78	0.9115	7.33	33.78	0.9115	7.29
	Q=36	32.91	0.892	**7.34**	32.91	0.892	7.37	32.91	0.892	7.38	32.91	0.892	7.34
	Q=42	32.28	0.8734	**7.40**	32.28	0.8734	7.43	32.28	0.8734	7.44	32.28	0.8734	7.40

From these figures and table it can be clearly observed that in all videos the set T_1=300 and T_2=600 outperforms all the other threshold sets with high PSNR and low search points.

5.2.3 Comparison of HYBHKS algorithm with other ME Algorithms

From section 5.2.1. it is ascertained that the proposed HYBHKS algorithm is a better one. Hence in this section it is compared with reputed motion estimation algorithms like FS, KCDS, HEXS for various test videos setting quantization parameter Q is set to 24, 30, 36 and 42. The frame rate is set to 25kbps. The parameters average PSNR, SSIM, encoded bit rate, compression ratio and search points are calculated and tabulated in tables 5.5, 5.6, 5.7 and

5.8. PSNR and Encoded bit rate are determined for various Q values Q=24, 30, 36 and 42 and Rate- Distortion (R-D) curves are plotted in fig.5.7 (a)-(d).

Table 5. 6: Average PSNR, SSIM, Encoded Bit Rate, Compression Ratio (4:2:0 format) and Search Points of FS, KCDS, HEXS and HYBHKS for the test video Akiyo_cif

Q	ME Algorithm	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio	Search Points
Q=24	FS	41.28	0.949	651.79	46.6604	196.63
	KCDS	41.28	0.949	651.91	46.652	**7.96**
	HEXS	41.27	0.949	653.21	46.55925	9.42
	HYBHKS	41.28	0.949	651.91	**46.6521**	8.15
Q=30	FS	40.21	0.94	532.58	57.1043	196.63
	KCDS	40.21	0.94	532.55	57.1074	**8.50**
	HEXS	40.2	0.94	533.66	56.9888	9.55
	HYBHKS	40.21	0.94	532.58	57.1048	8.58
Q=36	FS	39.34	0.9295	468.37	64.9333	196.63
	KCDS	39.34	0.9295	467.58	65.0431	**8.88**
	HEXS	39.34	0.9294	468.44	64.9241	9.64
	HYBHKS	39.34	0.9295	467.57	**65.0438**	8.92
Q=42	FS	38.66	0.9208	417.17	72.902	196.63
	KCDS	38.67	0.9208	415.36	73.2207	9.36
	HEXS	38.66	0.9206	416.38	73.041	9.84
	HYBHKS	38.67	0.9207	415.45	73.2039	**9.31**

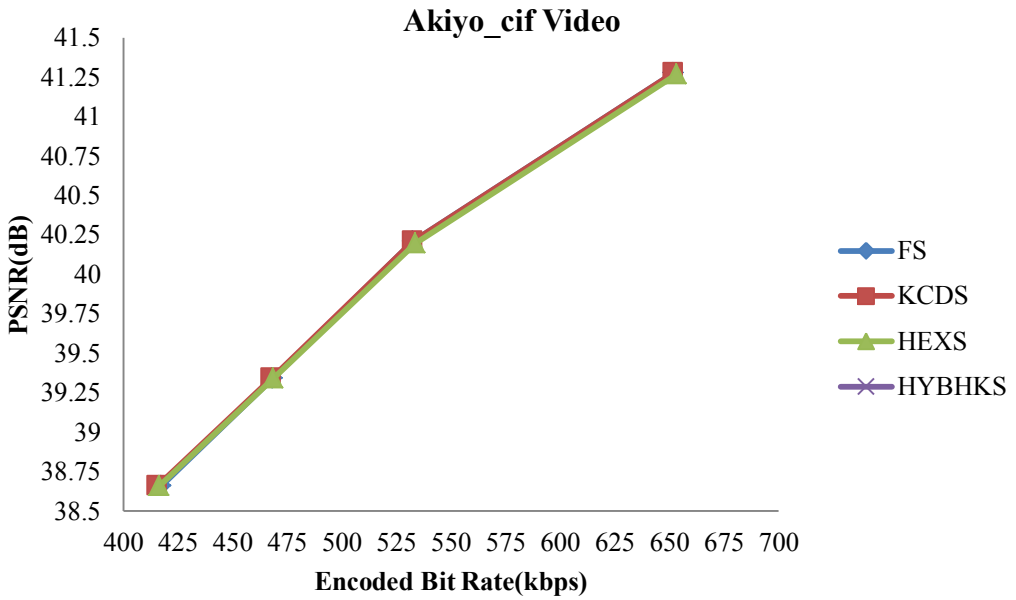

Fig.5.7. a: RD curve for the test video Akiyo_cif

Table 5. 7: Average PSNR, SSIM, Encoded Bit Rate, Compression Ratio (4:2:0 format) and Search Points of FS, KCDS, HEXS and HYBHKS for the test video Foreman_cif

Q	ME Algorithm	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio	Search Points
Q=24	FS	37.75	0.9035	1523.21	19.9663	196.63
	KCDS	37.74	0.9034	1533.34	19.8344	16.06
	HEXS	37.68	0.9023	1569.23	19.38065	14.05
	HYBHKS	37.73	0.9032	1519.01	**20.0215**	**10.086**
Q=30	FS	36.76	0.8851	1215.66	25.0174	196.63
	KCDS	36.75	0.8849	1226.35	24.7995	16.33
	HEXS	36.69	0.8835	1253.44	24.2634	14.21
	HYBHKS	36.73	0.8846	1210.94	**25.115**	**10.39**
Q=36	FS	36.04	0.9295	1012.04	30.0509	196.63
	KCDS	36.03	0.9295	1022.43	29.7457	16.60
	HEXS	35.98	0.9294	1044.45	29.1186	14.37
	HYBHKS	36.01	0.9295	1007.15	**30.1969**	**10.60**
Q=42	FS	35.51	0.8536	865.64	35.1333	196.63
	KCDS	35.49	0.8530	876.67	34.6912	16.87
	HEXS	35.44	0.8516	894.34	34.0058	14.48
	HYBHKS	35.47	0.8527	860.57	**35.3401**	**10.77**

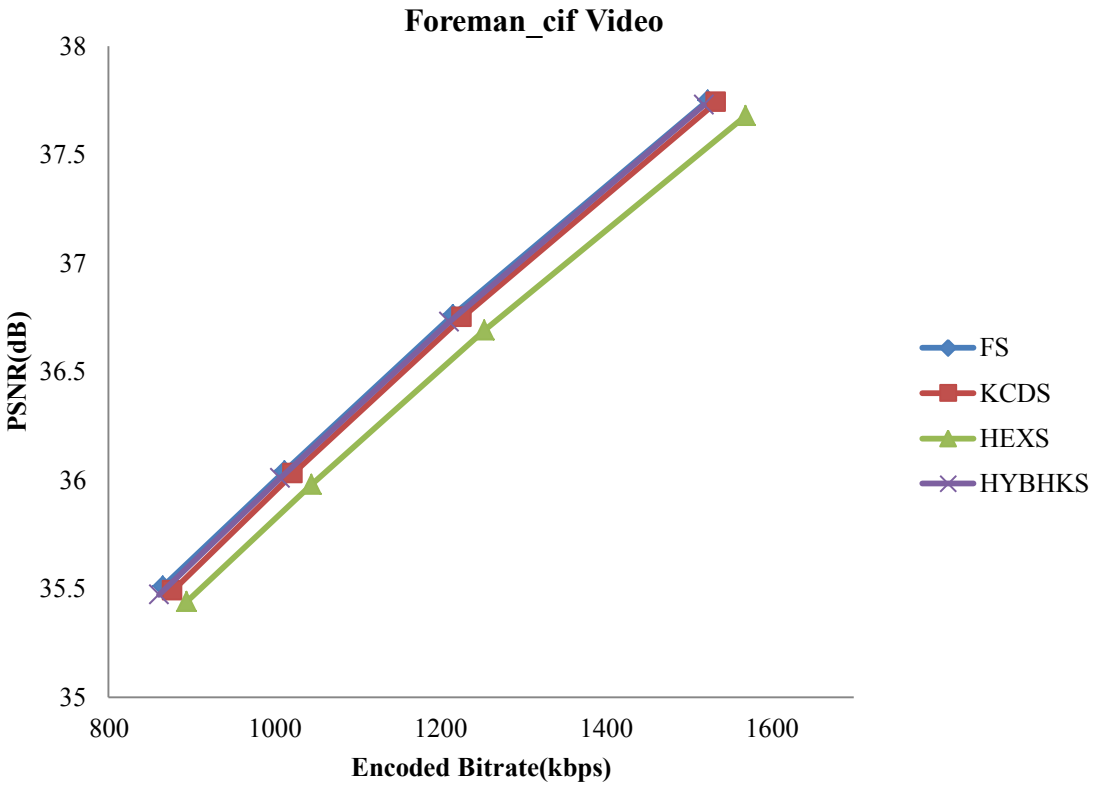

Fig.5.7. b: RD curve for the test video Foreman_cif

Table 5. 8: Average PSNR, SSIM, Encoded Bit Rate, Compression Ratio (4:2:0 format) and Search Points of FS, KCDS, HEXS and HYBHKS for the test video Ice_cif

Q	ME Algorithm	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio	Search Points
Q=24	FS	40.66	0.9508	883.94	34.4058	196.63
	KCDS	40.66	0.9507	886.29	34.31335	17.47
	HEXS	40.6	0.9502	913.24	33.30225	14.19
	HYBHKS	40.62	0.9504	895.96	33.94455	**11.10**
Q=30	FS	39.58	0.9422	733.59	41.4577	196.63
	KCDS	39.57	0.9421	736.44	41.2969	17.80
	HEXS	39.52	0.9413	758.28	40.1078	14.39
	HYBHKS	39.55	0.9416	744.18	40.8675	**11.74**
Q=36	FS	38.79	0.9342	634.08	47.9637	196.63
	KCDS	38.79	0.934	636.89	47.7519	18.02
	HEXS	38.73	0.9331	654.83	46.4439	14.45
	HYBHKS	38.76	0.9335	643.55	47.2575	**11.95**
Q=42	FS	38.21	0.9273	559.4	54.3664	196.63
	KCDS	38.2	0.9271	562.09	54.1064	18.13
	HEXS	38.15	0.9261	577.46	52.6669	14.49
	HYBHKS	38.18	0.9265	568.09	53.5356	**12.07**

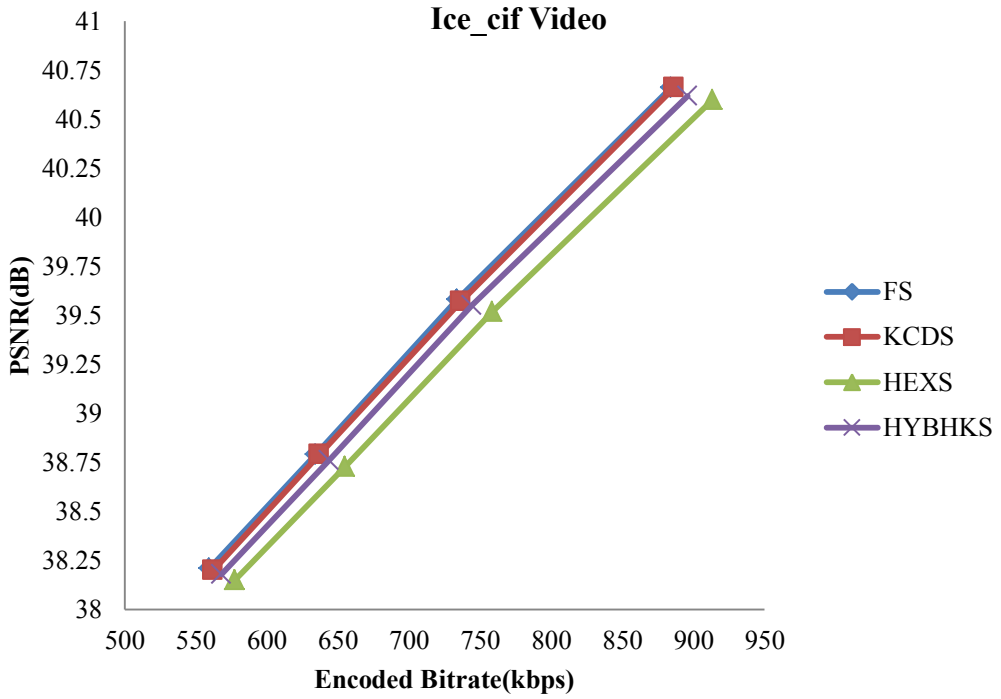

Fig.5.7. c: RD curve for the test video Ice_cif

Table 5. 9: Average PSNR, SSIM, Encoded Bit Rate, Compression Ratio (4:2:0 format) and Search Points of FS, KCDS, HEXS and HYBHKS for the test video Mobile_cif

Q	ME Algorithm	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio	Search Points
Q=24	FS	35.04	0.9314	4033.51	7.54	196.628788
	KCDS	35.04	0.9314	4033.5	7.54	7.531246
	HEXS	35.01	0.9309	4137.74	7.3501	8.399209
	HYBHKS	35.04	0.9314	4033.8	7.5395	**7.25729**
Q=30	FS	33.78	0.9114	3376.38	9.0075	196.628788
	KCDS	33.78	0.9115	3376.28	9.0078	7.549613
	HEXS	33.76	0.9108	3461.87	8.7851	8.7851
	HYBHKS	33.78	0.9115	3376.31	9.0077	**7.28979**
Q=36	FS	32.9	0.8919	2906.62	10.4633	196.628788
	KCDS	32.91	0.892	2906.77	10.4627	7.580244
	HEXS	32.88	0.8912	2980.83	10.2028	8.434604
	HYBHKS	32.91	0.892	2906.57	10.4635	**7.338401**
Q=42	FS	32.28	0.8734	2546.45	11.9432	196.628788
	KCDS	32.28	0.8734	2546.48	11.9431	7.591753
	HEXS	32.26	0.8725	2611.5	11.6457	8.435758
	HYBHKS	32.28	0.8734	2546.67	11.9422	**7.396254**

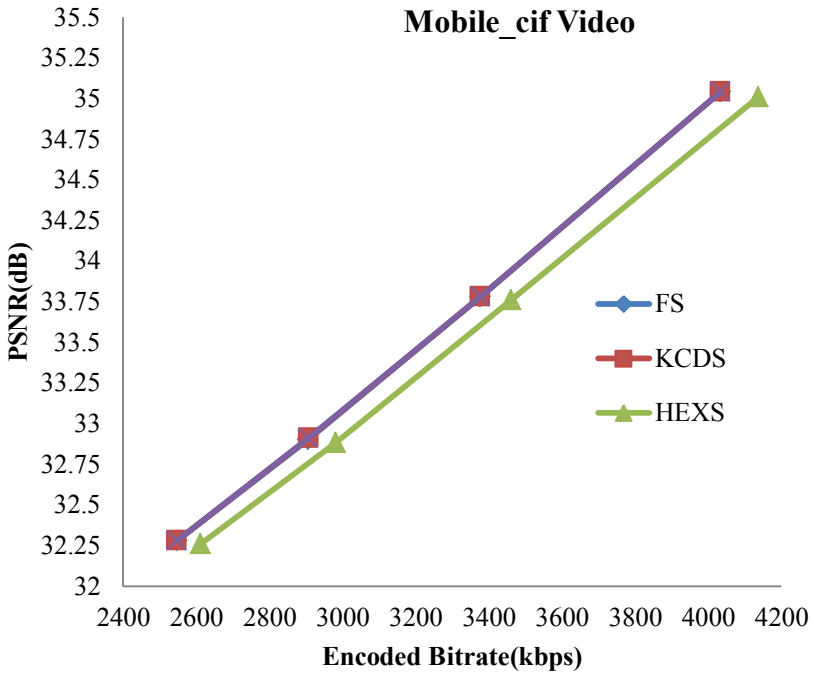

Fig.5.7. d: RD curve for the test video Mobile_cif

Fig.5. 7: RD curves

From the figures 5.7 (a)-(d) and the tables 5.5-5.8 depict that the proposed HYBHKS algorithm performs well with high PSNR and comparatively low encoded bit rate with reduced computational complexity. Fig.5.8 depicts the original and reconstructed frames (frame no.26) of the real time video using some ME algorithms mentioned. Fig.5.9 shows the original and reconstructed frame (frame no.10) of the video Akiyo_cif using all the ME algorithms discussed earlier. It is observed that the frame is reconstructed properly without compromising with the quality of the video using very few search points per macro-block.

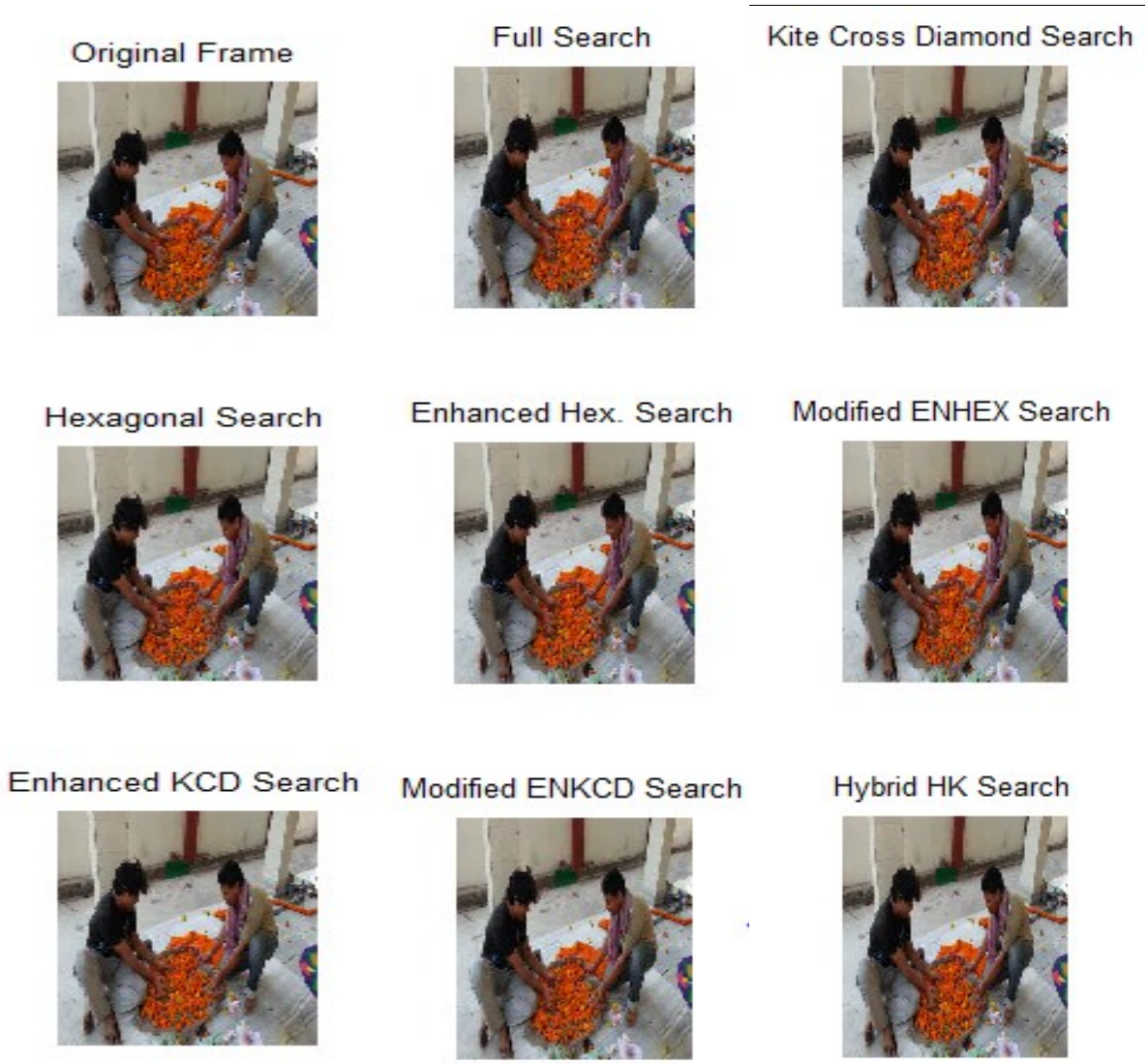

Fig.5. 8: Frame 26 and its corresponding reconstructed frames using ME algorithms

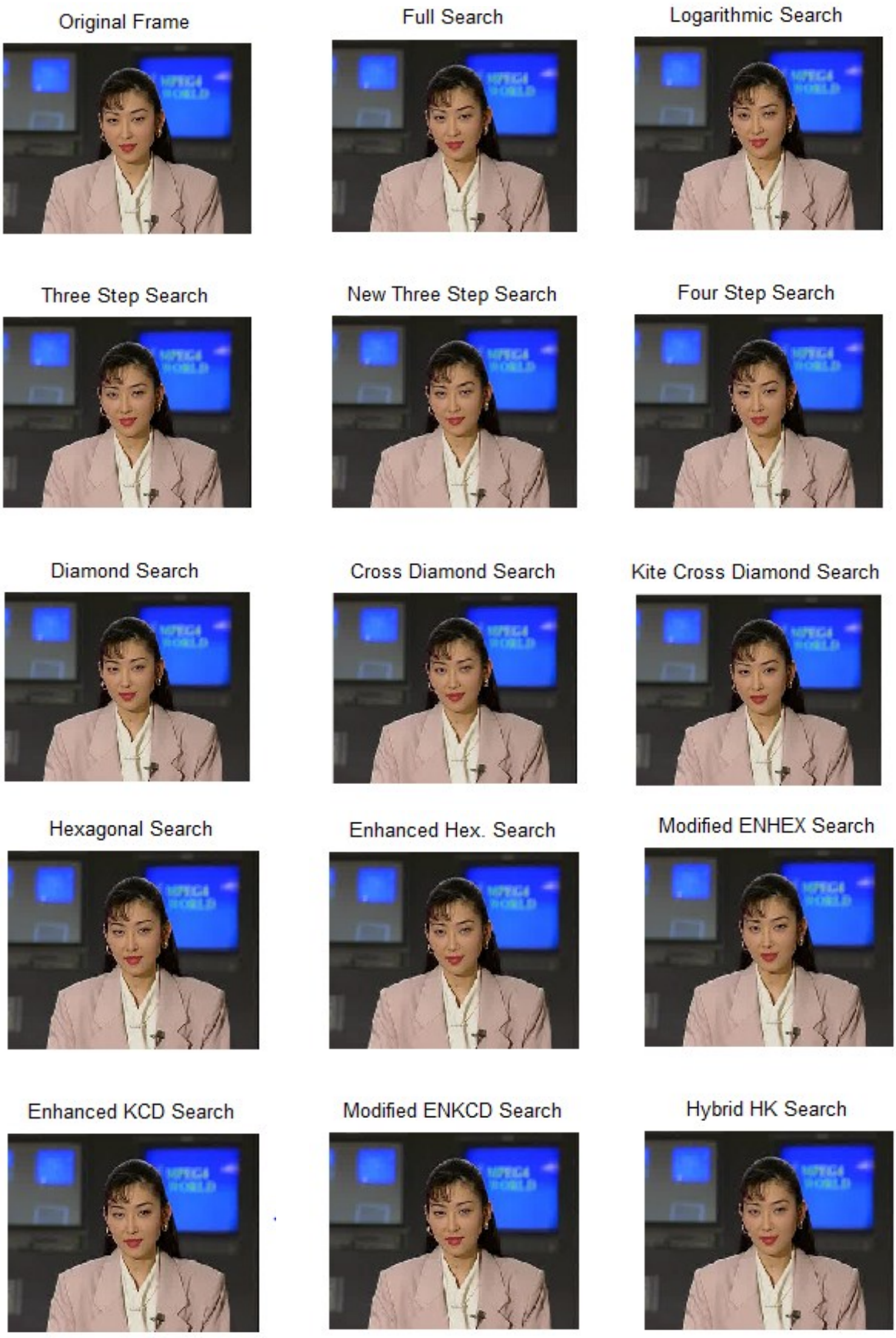

Fig.5. 9: Frame 10 of akiyo_cif video and it corresponding reconstructed frames using ME

CHAPTER 6

VIDEO COMPRESSION USING DISCRETE HARTLEY TRANSFORM

6.1 Introduction of Discrete Hartley Transform

As discussed in chapter 2 Discrete Hartley Transform can be used interchangeably with DCT in video encoding. In this chapter analysis is done on the video encoding process using DHT.

DHT expresses a finite sequence of data samples in terms of a sum of cosine and sine functions oscillating at different frequencies instead of only cosine terms [4]. The forward DHT transform coefficients of a block of pixels x(m,n) of size M×N using DHT is given by

$$X(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)\cos\left(2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right) + \sin\left(2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right) \qquad (6.1)$$

Where k=0,1,2,…M-1 and l=0,1,2,..N-1.

The inverse DHT coefficients are obtained by using the formula

$$\hat{x}(m,n) = \frac{1}{MN}\sum_{k=0}^{M-1}\sum_{l=0}^{N-1} X(k,l)\cos\left(2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right) + \sin\left(2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right) \qquad (6.2)$$

Where m=0,1,2,…M-1 and n=0,1,2,…N-1.

6.2 Energy Quantization

DHT coefficients don't follow zigzag scanning. It is very difficult to design the quantization matrix for DHT. Hence Energy Quantization is used for DHT.

In this technique first the normalised energy of the transformed coefficients is found out using the formula

$$E_n = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left(x(m,\ n)\right)^2 \qquad (6.3)$$

Where x is an input image of frame size M×N.

A percentage of the normalised energy is set as the threshold value for quantization. Then the energy of each pixel is compared to that threshold value. If the energy is greater than threshold then the transformed coefficient is kept as it is. Otherwise it is made to zero. Depending on the requirement of compression, the threshold value can be changed.

6.3 Scanning Order of the Quantized DHT coefficients

The scanning order should be such that the coefficients having high energy are scanned first followed by low energy coefficients. Analysing the kernel of DHT it can be observed that the coefficients equidistant from the origin across row and column are given equal priority. Hence the scanning order for DHT is as shown in fig.6.1.

1	2	4	6	8	10	12	14
3	16	17	19	21	23	25	27
5	18	29	30	32	34	36	38
7	20	31	40	41	43	45	47
9	22	33	42	49	50	52	54
11	24	35	44	51	56	57	59
13	26	37	46	53	58	61	62
15	28	39	48	55	60	63	64

Fig.6. 1: Scanning order of DHT [4]

6.4 Procedure of Video Encoding and Decoding of DHT Coefficients

The following procedure is used at the encoder to encode and decode the DHT coefficients:

At the Encoder:

1. First the frame (of I or P type) is divided into blocks of size 8×8.
2. DHT is performed on each block.
3. Energy Quantization is done.
4. The obtained coefficients are normalised by dividing with the Q-parameter.
5. The coefficients are scanned using the order mentioned in fig.6.1.
6. Then Run Level Encoding (RLE) is done followed by Huffman encoding.

At the Decoder:

1. The coefficients are decoded using the look up tables of Huffman Encoding and RLE.
2. The obtained coefficients are rearranged in blocks.
3. Then denormalization is done. (Dequantization cannot be done in DHT).

4. Inverse DHT is performed and the frame is reconstructed.

6.5 Results and Discussion

The bit stream of the encoded data should contain headers that convey the information regarding normalization parameter, frame rate, format of frame, frame size etc. It is a difficult task to design the bit stream. Hence in this analysis encoding is performed on main contents of bit stream like DHT coefficients, motion vectors and coded block pattern. 10% of the output encoded data size is allotted for headers. Only luminance component of the video is encoded.

6.5.1 Comparison of DCT and DHT for 8×8 block size

To compare the performance of DHT with DCT, the DCT bit stream is also encoded similar to that of DHT containing only DCT coefficients, motion vectors and coded block pattern. In this analysis, Diamond Search algorithm is used for Motion Estimation. Average PSNR, average encoding frame rate, average search points per macro block and compression ratio are calculated for the test videos Akiyo_cif, News_cif and Mobile_cif and tabulated in table 6.1.

Table 6. 1: Comparison of DCT and DHT using Average PSNR (dB), Average Frame Rate (fps), Average Search Points per macroblock and Compression Ratio

Test Video	Q	Transform & Threshold	PSNR (dB)	Frame Rate (fps)	Search Points	Compression Ratio
AKIYO	24	DCT	40.1063	0.8819	13.3405	30.692
	24	DHT, $0.8*E_n$	36.7178	0.7459	14.5663	35.5406
	30	DCT	39.6472	0.8479	13.7439	51.7034
	30	DHT, $0.8*E_n$	35.1589	0.7812	14.7381	53.0405
NEWS	24	DCT	35.9149	0.5354	12.8996	19.7136
	24	DHT, $0.8*E_n$	34.8277	0.5734	13.0364	22.4065
	30	DCT	34.6909	0.5590	13.0364	23.1212
	30	DHT, $0.8*E_n$	33.3476	0.606	17.1476	25.9135
MOBILE	24	DCT	31.7404	0.1644	11.6962	5.3586
	24	DHT, $0.8*E_n$	30.2071	0.1683	16.1417	5.3106
	30	DCT	29.6312	0.1988	12.4558	6.4766
	30	DHT, $0.8*E_n$	29.1306	0.1910	16.1798	5.9794

From table 6.1, it is observed that for Q=24 DHT gives more compression ratio with a bit low PSNR. But as increasing from 24 to 30, DCT gives a better PSNR value for similar compression ratio. Also the search points of DCT are low when compared to that of DHT.

6.5.2 Comparison of DCT and DHT for various block sizes

Since DHT using 8×8 block size did not give better results than DCT, the analysis of DHT is extended to 16×16 blocks. In this analysis, each frame is divided into blocks each of size 16×16. Each block is transformed using 16×16 DHT kernel. Then energy quantization is done followed by normalisation. The scanning order type for DHT in fig.6.1 is analysed. The coefficients in the first row and first column are scanned alternately followed by second row and second column and so on. The same pattern is extended to 16×16 block size as shown in fig.6.2.

1	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30
3	32	34	.	.	.	.	.	.							59
5	33	61													.
7	35		.												.
9	.			.											.
11	.				.										.
13	.					.									.
15	.														
17	.														
19															
21															
23															.
25												.			.
27													.		.
29														.	254
31	60	.	.	.							.	.	.	255	256

Fig.6. 2: Scanning Order of 16×16 DHT

Then the scanned ones are run-level coded and Huffman encoded. At the decoder the vice versa is implemented.

The comparison is done for the test videos Akiyo_cif and Foreman_cif using Diamond Search motion estimation algorithm, Q=30 and the results are tabulated in the table

6.2. The threshold for DHT is set to $0.6*E_n$. DCT for 16×16 is not done since the quantization matrix in DCT is present only for 8×8 size and extending it to 16×16 is a difficult and time taking task.

Table 6. 2: Comparison of DCT and DHT for various block sizes (Q=30)

Test Video	Transform	PSNR (dB)	SSIM	Frame Rate	Search Points	Compression Ratio
AKIYO	DCT 8×8	**39.64718**	**0.9305575**	0.847944	**13.7439**	**51.70341412**
	DHT 8×8	38.4036	0.90687899	0.768328	14.7381	33.75027294
	DHT 16×16	38.02814	0.88983724	**0.951769**	14.63071	42.37924968
FOREMAN	DCT 8×8	**34.83255**	**0.8010273**	0.327425	25.47586	**11.49483589**
	DHT 8×8	34.44585	0.78659903	0.317254	25.6184	9.708243118
	DHT 16×16	34.29759	0.77714041	**0.399106**	**25.4283**	10.82265889

From the above results, it is observed that the average PSNR and average SSIM are more for DHT 8×8 block rather than DHT 16×16. Although the compression ratio is large for 16×16 DHT than 8×8 DHT, it is less when compared to 8×8 DCT. Hence from the above results it is concluded that DCT 8×8 transformation is better than DHT transform.

CHAPTER 7

VIDEO COMPRESSION USING DECIMATION TECHNIQUES

A real time video consists of many objects that are spread along horizontal and vertical directions. The properties of the whole object will be more or less similar. Resizing the high resolution frames to low resolution ones cannot make huge information to lose. Transmitting this low resolution video can make us attain more compression ratio. Also it will reduce the time to encode. The received low resolution video can be up-sampled to original size using interpolation techniques.

7.1 Compression using alternate rows

Each row in a video are highly correlated to its neighbouring rows in spatial and temporal domains. Hence one can skip the alternate rows in a frame that can be easily reconstructed using spatial and temporal correlation properties of the rows in real time videos. It can also increase the compression ratio of the video encoder. The two techniques are described in the following subsections.

7.1.1 Video Compression using Spatial Correlation of rows

In this technique the property of correlation of consecutive rows in a frame is considered. Hence only one row in two consecutive rows is transmitted. At the decoder the skipped row is reconstructed using spatial averaging of its consecutive preceding and succeeding rows. At the encoder the pattern depicted in fig.7.1 is followed to attain low resolution video. The alternate rows in each frame are skipped to have a low resolution video. Then it is fed as the input video to the video encoder. Encoding is done as described in the chapter 2 and bit stream is generated. Then it is transmitted to the receiver. At the receiver the data is decoded using the reverse procedure followed at the video encoder and low resolution video is reconstructed. Now the low resolution video is up-sampled to original size by reconstructing the skipped rows by averaging the neighbouring rows of the corresponding frames as depicted in fig.7.1.

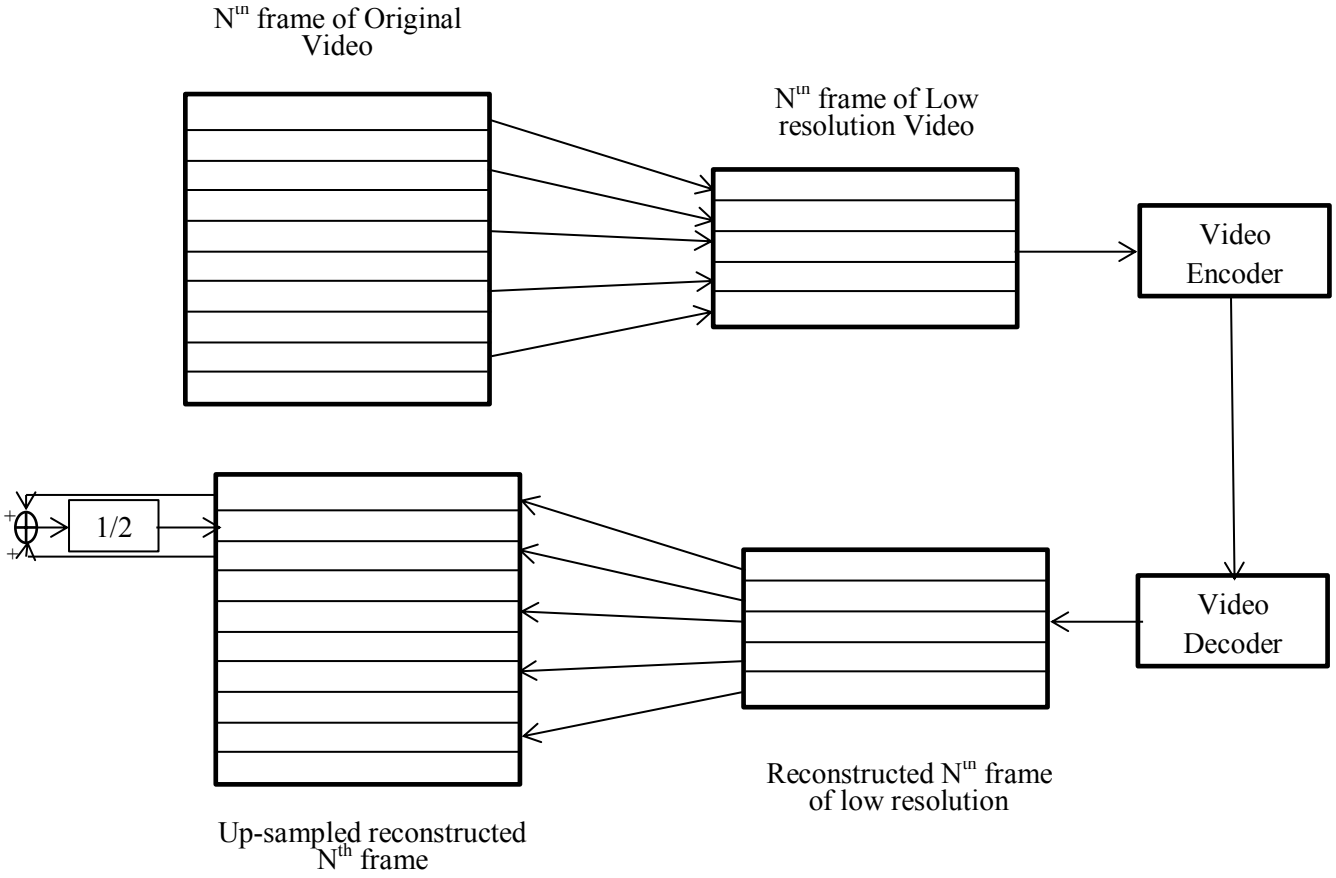

Fig.7. 1: Video compression using spatial correlation of rows

7.1.2 Video Compression using Temporal Correlation of rows

In this technique the property of correlation of rows in consecutive frames is considered. Hence alternate rows in two consecutive frames are transmitted to the encoder. At the decoder the skipped row is reconstructed using temporal averaging of its consecutive rows in preceding and succeeding frames. At the encoder the pattern depicted in fig.7.2 (a) is followed for odd frames whereas the pattern in fig. 7.2 (b) is followed for even frames.

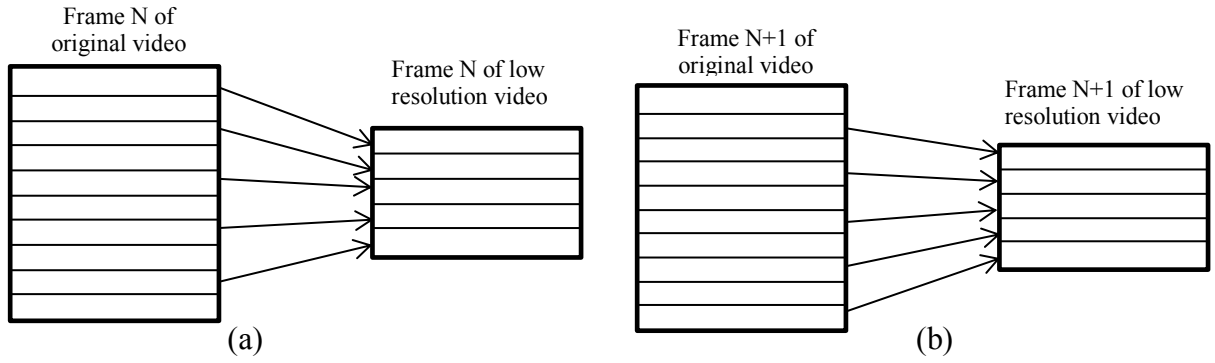

Fig.7. 2: Pattern used for converting original video to low resolution video

Then the obtained low resolution video is encoded and transmitted. At the receiver the bit stream is decoded and the low resolution video is reconstructed. Then it is up-sampled to original dimensions using the pattern shown in fig.7.3.

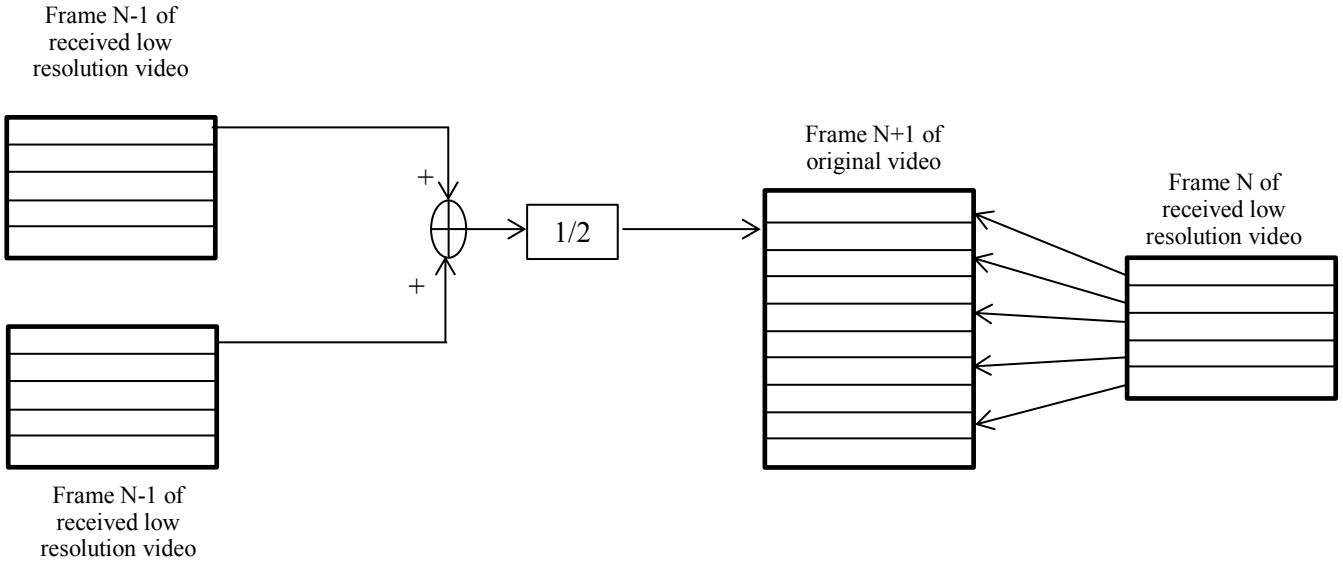

Fig.7. 3: Pattern used for up-sampling the low resolution video

7.2 Compression using Down Sampled Video

Compression Ratio of the video can be further increased by reducing the size of the video to half of its dimensions. The attained down sampled video is encoded and transmitted. At the decoder the received video is up-sampled to its original size using interpolation techniques like bicubic interpolation, lanczos3 interpolation and DCT based interpolation. The process is depicted in fig.7.4.

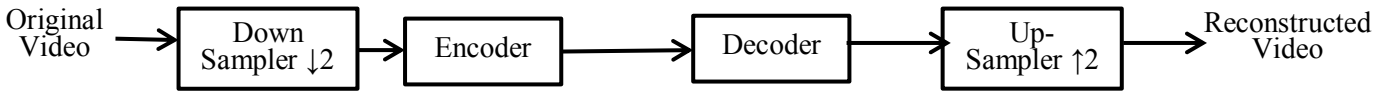

Fig.7. 4: Encoding and decoding of down sampled video

Bicubic and Lanczos3 interpolations are already inbuilt in Matlab. In DCT based interpolation [5], the frame is converted to DCT domain and resized to original dimensions by zero padding. Then the padded image is converted to spatial domain using IDCT.

7.3 Enhancement of the Received Video

In the above technique only 25% of the total video is transmitted. Reconstructing the video from this transmitted information may cause the output videos to blurr mainly at the edges. Hence the quality of the output video is degraded [8]. There is a need to enhance quality of the received video.

7.3.1 Using Adaptive Unsharp Masking

This technique [5] uses the concept of unsharp masking to reduce the blurr that occur at the edges. Generally low pass filtering an image gives the blurred image. A mask containing the information of edges can be attained by subtracting the original image with low pass filtered output. The edges of the original image are boosted by adding the weighted portion of the mask to the original image. Now when we up-sample this sharpened image the edges are not blurred that much resulting in a better quality high resolution image.

In Adaptive Unsharp Masking, the mask first frame of the decoded video is extracted. Then the optimum weight k_{opt} is selected adaptively which gives the maximum PSNR. Using this k_{opt} value the whole video is sharpened and then up-sampled. The process is depicted in fig.7.5.

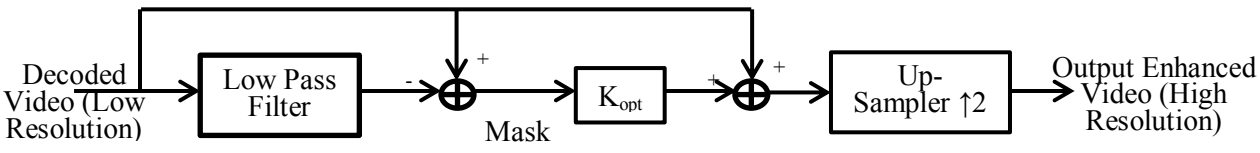

Fig.7. 5: Adaptive Unsharp Masking

7.3.2 Using proposed DWT Edge Boosting

Discrete Wavelet Transform decomposes an image into four frequency bands LL, LH, Hl and HH as shown in fig.2. LL is the original image low pass filtered and sub-sampled in horizontal and vertical directions. LH, HL and HH contain the residual horizontal, vertical and diagonal frequencies respectively.

During interpolation of an image the blurring of diagonal edges will be dominant when compared to that of horizontal and vertical edges. The proposed DWT edge boosting is concentrated on sharpening diagonal edges more.

In this method, a weight k ranging between 0 and 2 is multiplied to HH component of the DWT transformed image. Then it is inverse DWT transformed in order to get an edge

boosted image. Then up-sampling is done using interpolation techniques. The process is depicted in fig.7.6.

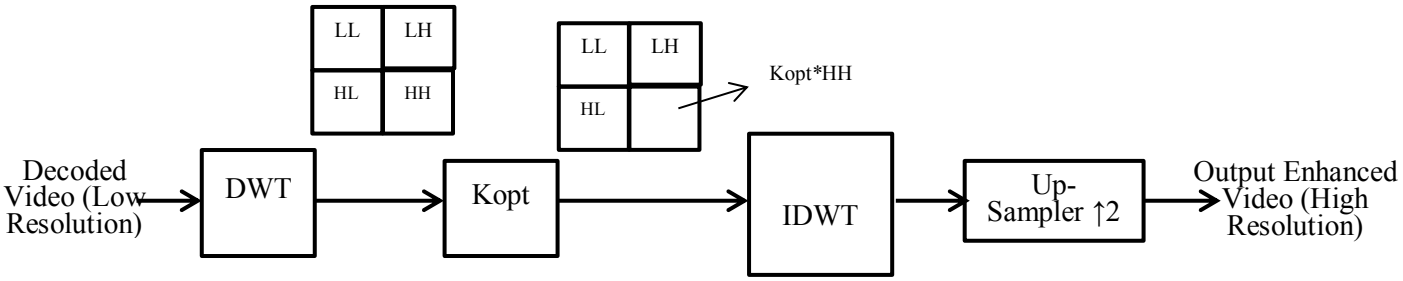

Fig.7. 6: Up-sampling using DWT Edge Boosting

7.4 Results and Discussion

7.4.1 Comparison of compression using original video and down sampled video using alternate rows

The alternate rows of test videos are skipped as discussed in the chapter 7.1 and the low resolution video is fed to video encoder. The quantization parameter is set to 24 and HYBHKS motion estimation algorithm is used for motion estimation. Test videos Akiyo_cif, Foreman_cif and Ice_cif are used. The output video is up-sampled to original size using spatial and temporal correlations. Table 7.1 presents the performance parameters without down sampling whereas tables 7.2 and 7.3 present the results for the video compression using spatial and temporal correlations respectively. Fig.7.7 and fig.7.8 depict frame 10 and its processing output frames of foreman_cif video at each step for the techniques using spatial and temporal correlations respectively. Fig.7.9 and fig.7.10 depict the same for Ice_cif video.

Table 7. 1: Performance parameters using video encoding using original video

Test Video	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio (4:2:0 format)	O/P Video Size (in bits)	Search Points
Akiyo	41.28	0.949	651.91	46.6521	7822880	8.153519
Foreman	37.73	0.9032	1519.01	20.0215	18228072	10.085976
Ice	40.62	0.9504	895.96	33.94455	8601168	11.101852

Table 7. 2: Performance parameters using compression using spatial correlation of rows

Test Video	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio (4:2:0 format)	O/P Video Size (in bits)	Search Points
Akiyo	37.7456	0.9073	453.58	67.05	5443008	7.544175
Foreman	33.7457	0.7792	986.75	31.1703	11708368	9.318485
Ice	36.6313	0.896	619.29	49.1093	5945160	10.298043

Table 7. 3: Performance parameters using compression using temporal correlation of rows

Test Video	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio (4:2:0 format)	O/P Video Size (in bits)	Search Points
Akiyo	37.2095	0.8836	587.45	51.7712	7049360	9.450572
Foreman	32.7499	0.6843	1067.28	28.4956	12807368	9.755488
Ice	35.4726	0.8279	687.73	44.2218	6602240	11.173359

Comparing the results of table 7.2 and 7.3 with that of table 7.1, it is observed that the compression technique using the spatial correlation of rows works well when compared to the technique using temporal correlation of rows in all aspects. It has maintained pretty high PSNR, SSIM and compression ratio values while attaining low encoded bit rate, output video size and search points which are the desired aspects in any compression scheme.

Fig.7. 7: Frame 10 of Foreman_cif video and its resized, decoded and reconstructed frames using spatial correlation of rows

Fig.7. 8: Frame 10 of Foreman_cif video and its resized, decoded and reconstructed frames using temporal correlation of rows

Fig.7. 9: Frame 10 of Ice_cif video and its resized, decoded and reconstructed frames using spatial correlation of rows

Fig.7. 10: : Frame 10 of Ice_cif video and its resized, decoded and reconstructed frames using temporal correlation of rows

From fig.7.6 and fig.7.7 it is easily noticed that the reconstructed frame using spatial correlation is good in terms of quality that the reconstructed frame using temporal correlation.

The latter is a bit hazy. The similar conclusions can be drawn by observing fig.7.9 and fig.7.10 of ice_cif video. Hence it is concluded that the compression scheme using the spatial correlation of rows is a satisfactory scheme.

7.4.2 Comparison of compression using down sampling followed by Up-sampling using interpolation, Unsharp Masking and DWT based Edge Boosting

The test videos are down sampled as described in chapter 7.2. and the low resolution videos are fed to video encoder. The quantization parameter is set to 24 and HYBHKS motion estimation algorithm is used for motion estimation. Test videos Akiyo_cif, Foreman_cif and Ice_cif are used. The output video is up-sampled to original size using interpolation, unsharp masking and DWT based edge boosting. Table 7.4 presents the performance parameters with down sampling and up-sampling whereas tables 7.5 and 7.6 present the comparison of quality parameters (PSNR and SSIM) in interpolating the video with unsharp masking and DWT edge boosting with simple interpolation technique respectively. Fig.7.10 and fig.7.12 depict the down sampled frame, its decoded frame and its upsmpled versions using various techniques for Akiyo_cif and Ice_cif videos respectively.

Table 7. 4: Performance parameters using video encoding of down sampled video followed by simple upsampling at the decoder

Test Video	PSNR (dB)	SSIM	Encoded Bit Rate (kbps)	Compression Ratio (4:2:0 format)	O/P Video Size (in bits)	Search Points
Akiyo	31.1968	0.7763	261.35	116.3674	3136216	6.355993
Foreman	27.7237	0.7352	466.12	65.2472	5593400	8.501582
Ice	29.1997	0.818	366.94	82.882	3522632	9.28931

Table 7. 5: Comparison of Video Quality Enhancement using Unsharp masking before Video Up-sampling

Test Video	Without Unsharp Masking		With Unsharp Masking					
			Lanczos3 Upsampling		Bicubic Upsampling		DCT based Upsampling	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Akiyo	31.1968	0.7763	31.2838	0.871	**31.2968**	**0.877**	31.1624	0.8607
Foreman	27.7237	0.7352	27.7801	0.7633	**27.8238**	**0.7694**	27.7479	0.7551
Ice	29.1997	0.818	30.0588	0.8604	**30.1689**	**0.8683**	29.872	0.8482

Table 7. 6: Comparison of Video Quality Enhancement using proposed DWT based Edge Boosting before Video Up-sampling

Test Video	Without Edge Boosting		With DWT based Edge Boosting					
			Lanczos3 Upsampling		Bicubic Upsampling		DCT based Upsampling	
	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM	PSNR (dB)	SSIM
Akiyo	31.1968	0.7763	31.3017	0.873	**31.3043**	**0.8775**	31.1928	0.8623
Foreman	27.7237	0.7352	27.8004	0.765	**27.8313**	**0.7702**	27.779	0.7571
Ice	29.1997	0.818	30.0982	0.8626	**30.1895**	**0.8698**	29.9266	0.8509

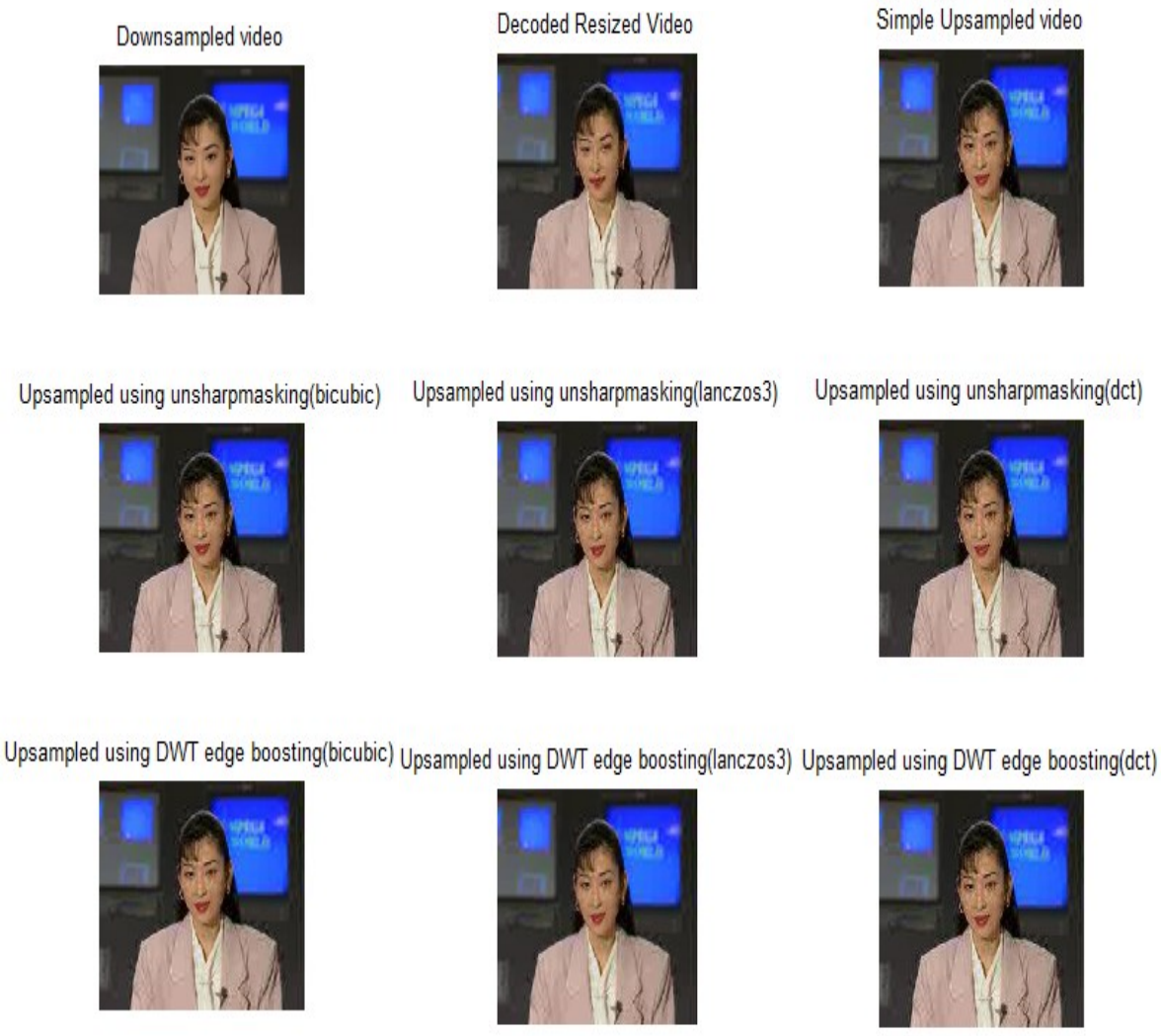

Fig.7. 11: Downsampled frame, its decoded and corresponding upsampled frames of frame 10 of Akiyo_cif video

Fig.7. 12: Downsampled frame, its decoded and corresponding upsampled frames of frame 10 of Ice_cif video

From table 7.4 the compression ratios are very high. But it is observed that PSNR and SSIM of output videos are low. In order to enhance the quality of videos unsharp masking is used to reduce the blur at the edges and up-sampled using lanczos3, bicubic and DCT based interpolation and results are tabulated in table 7.5. The quality of the video is improved a bit. Table 7.6 presents the quality parameters obtained using DWT based edge boosting followed by up-sampling. Comparing the results with that of unsharp masking it is noticed that the proposed Dwt based edge boosting has performed well in enhancing the quality of the video.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

Every electronic gadget in this techno-leading world has many more applications that use images and videos. Each and every communication between two people cannot be imaginable without images and videos. The limited bandwidth for transmission and limited memory make video compression a serious phenomenon to consider in the field of communications. There is a need to improve the video encoding process which can encode the video data with low computational complexity and high compression ratio along with maintaining quality quickly. The efficient encoding process is the one that is capable of removing almost all the redundancies in the video. Motion Estimation is the widely used scheme in many encoders like MPEG-2, MPEG-4 and H.264 that removes temporal redundancy. Full Search is the outdated and complex algorithm for ME. There is a wide scope of research in this particular area to reach out for the improved version of video encoder. Also the properties of each gadget differ from one another that limit the image compatibility. Videos with HDTV resolution cannot be played in normal mobiles. The vice versa will not be of visually acceptable quality. Hence there is a need to have good down sampling and up-sampling techniques.

In this thesis, many ME algorithms like Full Search, Logarithmic Search, Three Step search, Diamond Search, Kite Cross Diamond Search, Hexagonal Search, Enhanced Hexagonal Search are implemented and analysed. A few modifications to KCDS and Hexagonal Search are proposed. A novel Hybrid Hexagonal Kite Cross Diamond Search algorithm is proposed in this research. This method is very advantageous in reducing the computational complexity in the videos containing different types of motion. It results in improving the encoding speed. The compression ratio is also good maintaining the quality of the video. Analysis is done on DHT and compared with DCT. It is concluded that DCT performs better than DHT.

Also the compression techniques involving the concepts of spatial correlation and temporal correlation of rows were implemented. It is concluded that the technique using spatial correlation of rows performs better than the technique with temporal correlation of

rows increasing almost 50% compression ratio than encoding the whole video. Analysis is done on down sampling and up-sampling of video before and after encoder and decoder respectively. It is observed that the compression ratio attained by encoding the down-sampled video is increased to 150% of the original one. But the quality of the video is degraded much at the edges. Hence unsharp masking is applied to the decoded video before up-sampling it. A new technique for edge boosting using DWT is proposed in this research. It uses the HH component of DWT in edge boosting the decoded video. Then it is up-sampled. The proposed technique is compared with unsharp masking and the conclusion that the performance the former is better than the latter is drawn.

8.2 Future Work

The motion estimation algorithms in this thesis are based on the cross-centre biased characteristic of the motion of objects which assumes that the probability of the best match of the block is more for its consecutive neighbouring pixels. This reduces the complexity of the estimator. Future work can be done on imparting the knowledge of motion detection to the estimator that may still reduce the computational complexity of the estimator. The down sampled and up-sampled videos have some blur content. Analysis can be done on enhancement techniques that may further increase the quality of the up-sampled video.

PUBLICATION

Narapaneni Ragasudha, Deepak Singh and Sukadev Meher, **"Block Based Motion Estimation Using Hybrid Hexagonal Kite Cross Diamond Search Algorithm,"** *IEEE International Conference on Communication and Signal Processing 2014,* Adiparasakthi College of Engineering, Melmaruvathur, April 2014.

REFERENCES

[1] "H.262;Transmission of Non-Telephone Signals; Information Technology- Generic coding of moving pictures and associated audio information: Video," ITU-T , 1996.

[2] M. Jakubowski and G. Pastuszak, "Block-based motion estimation algorithms—a survey," *Opto-Electronics Review,* vol. 21, no. 1, pp. 86-102, 2013.

[3] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Electronic Imaging,* volume 4671, pp.123-133, January 2002.

[4] D. Patel, A. Menoth Jose, N. Mascarenhas and S. Stany Monis, "JPEG Image Compression using DCT and DHT and Comparison of Both Techniques based on Mean Square Error and Peak Signal to Noise Ratio," *International Journal of Computer Applications,* vol. 81, no. 15, pp. 23-27, 2013.

[5] A. Acharya and S. Meher, "Region Adaptive, Unsharp Masking Based Lanczos-3 Interpolation for 2-D Up-Sampling: Crisp-Rule Versus Fuzzy-Rule Based Approach," in *Sensing Technology: Current Status and Future Trends II*, Springer, 2014, pp. 47-73.

[6] R. C. Gonzalez and R. E. Woods, "Digital image processing, 2nd," *SL: Prentice Hall,* 2002.

[7] I. E. Richardson, "Video Codec Design: Developing Image and Video Compression Systems, John Wiley & Sons Ltd., London,2009".

[8] A. Bovik, "The Essential Guide to Video Processing", London: Elsevier, 2009.

[9] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on,* vol. 13, no. 4, pp. 600-612, 2004.

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

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

[12] C.-H. Cheung and L.-M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 12, no. 12, pp. 1168-1177, 2002.

[13] C.-W. Lam, L.-M. Po and C. H. Cheung, "A novel kite-cross-diamond search algorithm

for fast block matching motion estimation," in *Circuits and Systems, ISCAS'04. Proceedings of the 2004 International Symposium on*, vol.3, pp.729-732, April 2004.

[14] C. Zhu, X. Lin, L.-P. Chau, K.-P. Lim, H.-A. Ang and C.-Y. Ong, "A novel hexagon-based search algorithm for fast block motion estimation," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, pp.1593-1596,2001.

[15] C.-Y. Su, Y.-P. Hsu and C.-T. Chang, "Efficient hexagonal inner search for fast motion estimation," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol.1, pp.1093-1096, May 2005.