Some Intra-Frame and Inter-Frame Processing Schemes for Efficient Video Compression

A Thesis Submitted In Partial Fulfillment Of The Requirements For The Degree Of

Master of Technology

in

Signal & Image Processing

by

Sobhan Kanti Dhara

213EC6259

Department of Electronics and Communication in Engineering

National Institute of Technology Rourkela

Odisha,India-769008

June,2015

Some Intra-Frame and Inter-Frame Processing Schemes for Efficient Video Compression

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Technology

in

Signal & Image Processing

by

Sobhan Kanti Dhara

213EC6259

Under the Supervision of

Prof. Sukadev Meher

Department of Electronics and Communication in Engineering

National Institute of Technology Rourkela

Odisha,India-769008

June,2015

Dedicated to My Family...

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA
ROURKELA, ODISHA, INDIA-769008

CERTIFICATE

This is to certify that the thesis titled, **"Some Intra-Frame and Inter-Frame Processing Schemes for Efficient Video Compression"**, submitted by **Mr. Sobhan Kanti Dhara** bearing **Roll No. 213EC6259** in partial fulfillment of the requirements for the award of the degree of **Master of Technology** in **Electronics and Communication Engineering** with specialization in **"Signal & Image Processing"** during session 2014-2015 at National Institute of Technology Rourkela is an original research work carried out under my supervision and guidance.

Prof. Sukadev Meher

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA
ROURKELA, ODISHA, INDIA-769008

DECLARATION

I certify that

1. The work contained in the thesis is original and has been done by myself under the supervision of my supervisor.
2. The work has not been submitted to any other Institute for any degree or diploma.
3. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Sobhan Kanti Dhara

Acknowledgment

I would like to express my gratitude to my thesis guide **Prof. Sukadev Meher** for his guidance, advice and support throughout my thesis work. I am especially indebted to him for teaching me both research and writing skills, which have been proven beneficial for my current research and future career. Without his endless efforts, knowledge, patience, and answers to my numerous questions, this research would have never been possible. The experimental methods and results presented in this thesis have been influenced by him in one way or the other. It has been a great honour and pleasure for me to do research under supervision of Prof. Sukadev Meher. Working with him has been a great experience. I would like to thank him for being my advisor here at National Institute of Technology, Rourkela.

Next, I want to express my respects to **Prof. K.K Mahapatra, Prof. S. K. Patra, Prof. Samit Ari, Prof. Manish Okade, Prof. A. K. Sahoo, Prof. L.P.Roy, Prof .A.K. Swain, Prof. D.P. Acharya, Prof. S. Maiti** for teaching me and also helping me how to learn. They have been great sources of inspiration to me and I thank them from the bottom of my heart.

I would like to thank to all my faculty members and staff of the Department of Electronics and Communication Engineering, N.I.T. Rourkela, for their generous help for the completion of this thesis.

I would like to thank all my friends my classmates, seniors and especially Ragasudha for thoughtful and mind stimulating discussions we had, which prompted to think beyond the obvious. I've enjoyed their companionship so much during my stay at NIT, Rourkela.

I am especially indebted to my parents and sister for their love, sacrifice, and support. My parents are my first teachers,and I am grateful to them for guiding my steps on the path of achievements since my infant hood.

Sobhan Kanti Dhara

Contents

Abstract

Rapid increase in digital applications due to recent advances in digital communication and devices needs significant video information storing, processing and transmitting. But the amount of original captured video data is huge and thus makes the system complex in all kind of video processing.But applications demand a faster transmission in different sized electronic devices with good quality.Along with, limited bandwidth and memory for storage makes it challenging. These practical constraints for processing a huge amount of video data, makes video compression as active and challenging field of research.

The aim of video compression is to remove redundancy of raw video while maintaining the quality and fidelity. For inter frame processing, motion estimation technique is significantly used to reduce temporal redundancy in almost all the video coding standards e.g. MPEG2, MPEG4, H264/AVC which uses state-of-art algorithm to provide higher compression with a perceptual quality.Though motion estimation is main contributor for higher compression, this is the most computationally complex part of video coding tools. So, it is always a requirement to design an algorithm that is both faster and accurate and provides higher compression but good quality output. The goal of this project is to propose an algorithm for motion estimation which will meet all the requirements and overcome all the practical limitations. In this thesis we analyze the motion of video sequences and some novel block matching based motion estimation algorithms are proposed to improve video coding efficiency in inter frame processing. Particle Swarm Optimization technique and Differential Evolutionary model is used for fast and accurate motion estimation and compensation. Spatial and temporal correlation is adapted for initial population. We followed some strategy for adaptive generations, particle population, particle location history preservation and exploitation. The experimental result shows that our proposed algorithm is efficient to maintain the accuracy. There is significant reduction of search points and thus computational complexity while achieving comparable performance in video coding.

Spatial domain redundancy is reduced skipping the irrelevant or spatially co-related data by different sub-sampling algorithm.The sub-sampled intra-frame is up-sampled at the receiver side. The up-sampled high resolution frame requires to have good qual-

ity . The existing up-sampling or interpolation techniques produce undesirable blurring and ringing artifacts. To alleviate this problem, a novel spatio-temporal pre-processing approach is proposed to improve the quality. The proposed method use low frequency DCT (Discrete cosine transform) component to sub-sample the frame at the transmitter side. In transmitter side a preprocessing method is proposed where the received sub-sampled frame is passed through a Wiener filter which uses its local statistics in 3×3 neighborhood to modify pixel values. The output of Wiener filter is added with optimized multiple of high frequency component. The output is then passed through a DCT block to up-sample. Result shows that the proposed method outperforms popularly used interpolation techniques in terms of quality measures.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Video applications are becoming indispensable part of today's communication. A large number of applications like video conferencing, video transmission, video streaming, remote monitoring, needs video transmission and storage.Electronic devices are scaled down and ever increasing demand of smart TV, smart phones, tablets, pads, phablets makes video application a prime requirement. But good quality video is always been the primary requirement for end user with a limitation of storage memory and bandwidth for transmission. Above all, maximum of all these applications is real-time and demand a faster transmission. This makes video compression an active area of research. The size of raw video is compressed for transmission and storage at transmitter side and decompressed in the receiver side with good quality display to the end users. There are so many video encoders present like MPEG 2, H.262, MPEG 4, H.264 etc. Fast and accurate motion compensation is a crucial task in all these standards of video coding. The maximum (70 % -80 %) overhead of all video coding standards is for motion estimation and compensation for high level compression.So, less computational complex motion compensation techniques are required which will maintain good quality along with good amount of compression. Motion compensation accomplish high compression ratio by reducing temporal redundancy but it can also be increased by reducing spatial redundancy too. There are so many sub-sampling methods are available in the literature which meets the require-

ment but it suffers from undesirable artefacts. So, a new technique is required which will reduce the spatial redundancy maintaining the quality.

1.2 Problem Statement

With the development of smart devices of real time applications, the demand for efficient video coding is highly required.It can be achieved by reducing temporal and spatial redundancy. As mentioned earlier motion compensation is used for temporal redundancy reduction and sub sampling is used for spatial redundancy reduction. This research is mainly aimed to manage the practical following practical constraints

1. In, all the standard video coding techniques, motion compensation adds the overall encoding complexity. The maximum overhead of computational complexity comes for this unit only but this is must for higher compression.

2. End user requirement with good quality and faster video transmission makes it difficult to design an accurate motion compensation algorithm.

3. Unnecessary artefact's, due to spatial redundancy removal also degrade the display quality.

1.3 Research Objective

The research aims at designing a computational less complex motion estimation algorithm for encoder to remove temporal redundancy. It also aims at proposing a novel algorithm to have good quality output after up-sampling of sub-sampled frame. The specific research objective is summarized as:

1. Design of generalized motion estimation algorithm which will be adaptive to the motion and will work efficiently with slow, medium and fast motion. The designed algorithm should be fast, accurate and less computationally complex and will give good quality output with higher compression.

2. Design of frame interpolation with less degree of blurring and ringing artifacts.

1.4 Thesis Organization

The thesis is organized as follows:

Chapter 2 is focused with the overview video encoding.

Chapter 3 describes the basics of motion estimation and existing algorithms on motion estimation .

Chapter 4 describes the detailed procedure of proposed motion algorithms with experimental results and analysis.

Chapter 5 is focused on the existing up-sampling algorithms and proposed method with the experimental results and discussion.

Chapter 6 conclude the research work done with and future scope

Chapter 2

Overview of Video Encoder

2.1 Introduction

This chapter gives some idea of basic video encoder. It briefly explains the overall process of basic video encoding technique. Then it moves into block based motion estimation and compensation which is main focus of the research and will be continued to the next chapter with details.

2.2 Overview

Video is nothing but a collection of frames and in real-life scenarios; all these frames are highly co-related to each other. So there are huge redundancies of the frames. These reductions of redundancy are main contributor to have compression[1]. It is found that two-dimensional frames suffer from following types of redundancies.

- **Coding redundancy:** The bit code used for intensity representation may contain more bits that it actually requires.
- **Spatial redundancy:** The pixels are spatially co-related as they have similar kind of information thus redundancy.
- **Temporal redundancy:** As video sequences are temporal orientation of frames, there are only few changes found in the consecutive frames.

- **Psycho visual redundancy:** Apart from above three there is another way to have the redundancy which is irrelevant information with respect to Human visual system (HVS). HVS cannot differentiate between less different intensity e.g. intensity 249 and 252 has same feeling to HVS.

So, compression can be achieved by removing all the above mentioned redundancy from the captured video. Compression can be classified by two types.

- **Lossless compression:** In lossless compression the decoded output don't suffers from any kind of loss in information.

- **Lossy compression:** Lossy compression loses information while decoded in the receiver side. But most of the compression strategy comes under lossy compression as it can give a compression up to 95% higher than lossless compression. As our human visual system (HVS) has certain persistence level and it fails to differentiate minute change in details, lossy compression technique utilize it and intentionally lose information to give higher compression but due to HVS, the perpetual quality remains almost same.

2.3 Basic Encoder

Video encoder processed the video frame in two ways:

- **Intra Frame Encoding:** This frame processing is purely based on current frame only rather any temporal related frame processing. This is easy, less complex having easy synchronization and self-decoding capability but has less compression.

- **Inter Frame Encoding:** Inter frames are expressed in terms of neighboring frame and encoded to remove temporal redundancy. This is little complex and give higher compression.

  The basic components of a video encoder are described as follows and shown in the Fig.2.2:

2.4 Block Transform

It is a linear and reversible transform used to map spatial domain each block or sub-image into transform domain data to reduce spatial redundancy. The commonly used transformations are Discrete Cosine Transform (DCT), Discrete Hartley Transform (DHT) and Discrete Wavelet Transform (DWT) which compact the energy.

1. **Discrete Cosine Transform (DCT):** Discrete Hartley Transform expresses finite sequence of samples in terms of a sum of cosine functions and sin transform oscillating at different frequencies. The transformed coefficients of pixels x(m,n) block of size M × N using DHT is given by

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

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

n=size of input data sample, x, y are co-ordinate location in spatial domain, u,v are co-ordinate location transformed domain respectively.

DCT removes the spatial correlation among the pixels and concentrate the maximum information near to the dc value or the origin i.e. (0, 0) position of the matrix. More we are away from the dc value, more the irreverent coefficient found.

2. **Discrete Hartley Transform (DHT):** Discrete Hartley Transform expresses finite sequence of samples in terms of a sum of cosine functions and sin transform oscillating at different frequencies. The transformed coefficients of pixels x(m,n) block of size M×N using DHT is given by

$$X(k,l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x(m,n)cos\left(2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right) + sin\left(2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)\right) \tag{2.2}$$

where k=0,1,2,. . . M-1 and l=0,1,2,..N-1.

3. **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 16 × 16 DWT is applied for this type of applications which performs better.

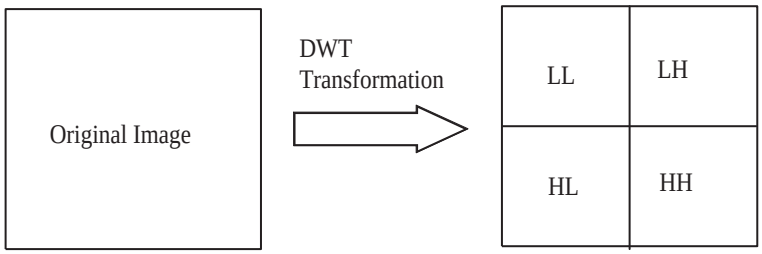


Figure 2.1: DWT Trasformation

DWT is a filtering operation which decomposes total image into four frequency bands LL, LH, HL and HH as shown in fig.2.2. LL is the low-pass filtered output of original image. LH, HL and HH is the residual horizontal, vertical and diagonal frequencies respectively. For further decomposition, LL part of previous output is further decomposed to again LL. LH, HL and HH.

Like DCT, in DWT most of the information lies in LL band and maximum pixel in LH, HL and HH band are zeros. Part of these zero value band can be removed and preserving relevant information and thus image or frame is compressed.

In DCT most of the information lies around origin i.e. (0, 0) position of the matrix and respectively irrelevant coefficients while traversing away from the origin. In DWT is that the memory requirement in computation is about half of required memory in DCT. In DWT, LL band has the most information rest of the band has maximum irrelevant data. So, we can see all the transform produce some irrelevant data which can be discarded for compression. But to reduce computational cost, we require this transform to work in block wise where DWT fails as it works on the whole frame. So, DCT is widely used in maximum video coding standards. But to reduce the computational cost block based DCT is used which produced unnecessary blocking artifacts.

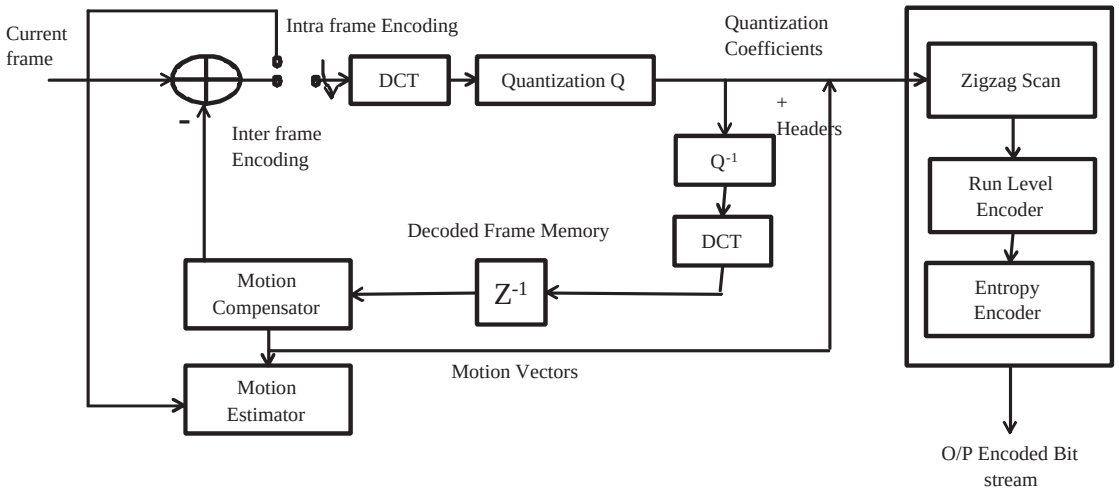


Figure 2.2: Block Diagram of Basic Video Encoder

2.5 Quantizer

This is lossy part in encoder and intentionally loses information to produce higher compression. As mentioned before HVS cannot differentiate between less different intensity,Quantization Matrix (QM) is designed accordingly which loses information keeping the perpetual quality almost intact taking the advantage of HVS. Apart from that,block-based video encoding schemes inherently loses information not only by removing truly redundant information but also by compromising quality intended to be minimally perceptible. Quantization Parameter (QP) is a parameter shows the degree of spatial detail saved. When For small QP,almost information is retained whereas for higher QP some information is lost and thus at the price of higher compression some quality is compromised. After transformation as remote AC coefficients almost irrelevant, it is quantized heavily rather than coefficients around DC value.

2.6 Symbol Coder

The quantized pixel value is encoded in this stage. It generates fixed or variable length codeword corresponding to each quantized output.

1. **Fixed Length Encoder:**In this coding, every generated code word has fixed length irrespective of the probability of occurrence.

2. **Variable Length Encoder:**Based on the probability of occurrence VLC generates variable length code word.It generates less probable symbol code word with long code and high probable symbol with short code word.

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

So, it is clear that VLC is more efficient to give high compression and thus used almost all video coding techniques. VLC consists of following module

2.6.1 Zig-Zag Scan

First two dimensional block transformed coefficients are arranged in one dimensional array in zigzag pattern. As mentioned before, due to DCT transformation, most of the information lies in DC coefficient and around it. So, zigzag scanning regroups the most informative low frequency components in top of the order shown in the figure 2.3.

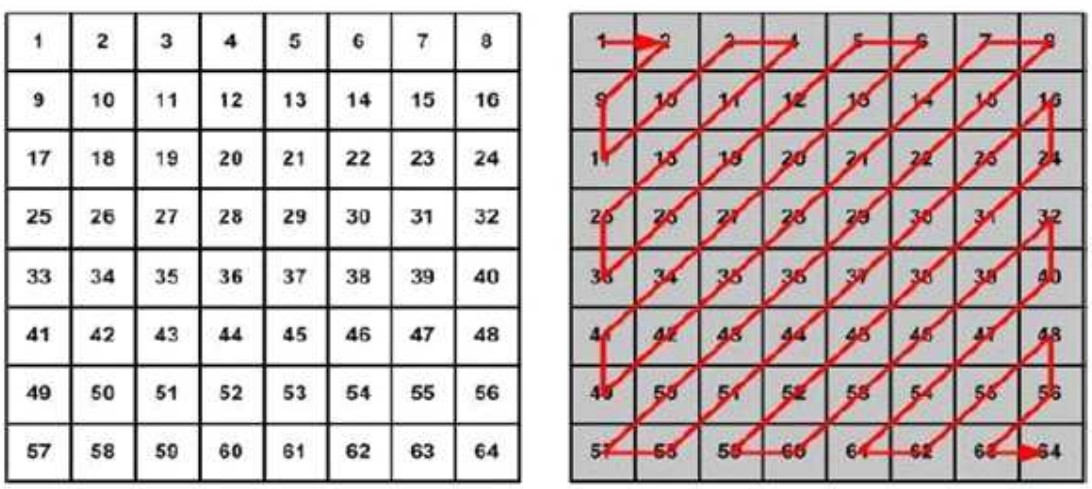


Figure 2.3: Zigzag Scanning

2.6.2 Run-Level Coding

Run-Level is coding technique where run-length of zeros followed by a nonzero level.The zigzag ordered output has most significant coefficients at top order whereas less significant coefficients. So there is a huge number of zero value coefficients in the zigzag. For efficient coding, in Run-Level coding,the length of the run is sent instead of sending these zeros. In run-level, run represents the number of zeros preceding the level which is a non-zero coefficient.

2.6.3 Entropy Encoder

The actually generates the bitstream. It represents the most probable occurring (run, level) to small code word whereas most probable occurring to respectively longer code word. The entropy encoder actually determines:

- Compression Efficiency
- Computational Efficiency and
- Error Robustness

Huffman coding is widely used entropy encoder. Huffman encoder generates the bit-stream based on the frequency of occurrence. It designs a table of variable length codes based on the probabilities and encode the levels. The decoder should have same table for decoding. Though, it is an efficient scheme in image but in video transmission, the table for each frame adds an additional overhead. It makes the system slow and even slower as it has to wait till the last to generate the table.So, video coding standard uses modified Huffman Coding to alleviate these obstacles.The decoder has this table for decoding.

2.7 Inverse Transformation and Inverse Quantization

The frame which is encoded through the above mentioned process is now decoded to provide reference to frame prediction for inter frame processing. A delay is kept for the same. The bit-stream is decoded and then it does the reverse processing in reverse order

to generate an approximation to the original frame. Though inverse transformation is an irreversible process and can exactly reconstruct the transformed output to original one but quantization followed by rounding off the data is responsible for loss of information.

2.8 Motion Estimation and Motion Compensation

This is a process of estimation of motion between consecutive frames to understand the spatial and temporal redundancy. In block based motion estimation, the frames are divided into some block and each block of current frame is checked within a search region of previous frame and match is checked based of some cost function of fitness function.The displacement of current block with best match block is the motion estimated and the displacement is known as motion vector. Motion Compensation technique generates the motion compensated predicted frame. The residue of the current frame and the motion compensated frame is encoded instead of total frame in inter frame processing.

Chapter 3

Motion Estimation

3.1 Introduction

The motion estimation and compensation is most important section in video codec mentioned already.An efficient motion estimation algorithm should have following key performance [2].

- **Coding Performance:** how much efficient algorithm is at generating minimized residual frame.
- **Computational complexity:** how easily the algorithm search the best minimized block
- **Side information:** how much additional side informations required to send to decoder
- **Error resilience:** how efficiently decoder is robust to the error occurred at transmission.
- **Scalability:** how efficient algorithm is irrespective search window size and type of motion
- **Quality Performance:** how much good quality output the algorithm can produce both qualitative and quantitative.

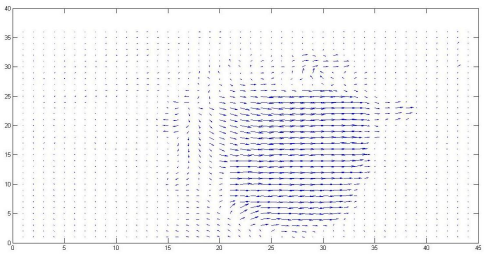
(a) Previous Frame

(b) Current Frame

(c) Motion Compensated Frame

(d) Residue without MC

(e) Residue with MC



(f) Motion Vector

Figure 3.1: Motion Compensation

- **Rate Distortion:** how efficient does it performing different compressed bit-rates.
- **Implementation:** Is the algorithm easily implementable both hardware and software.

Motion estimation creates a model based on the reference frames. This reference frame may be the past or future. The design goals for motion estimation is to model current frame based on the reference frame accurately whilst maintaining the all above key performance parameter mentioned above. Motion compensated residual is produced by subtraction of current frame with motion compensated frame. This is actually coded and sent to the receiver along with side information. This is motion compensated residual frame is reconstructed and stored for further prediction.The size of this residual frame is related to energy content in the frame and it is target of motion estimation and compensation to reduce the energy.Figure 3.1 shows motion vector and how motion compensation reduce the energy.

3.2 Block Matching

In any popular video coding standard e.g. MPEG-x, H.26x, VC-1 block matching techniques is used for motion estimation and compensation for its simplicity and ease of implementation. The frames are partitioned into non overlapping blocks of size 8×8 or 16×16. Then motion estimation algorithm searches block wise in neighboring area of previous frame where each block of the current frame is compared based on some similarity measures. The best match block gives the measures of motion and motion vector is calculated with the relative difference of the current block with the best match block of previous frame. The figure 3.2 shows the same. A video coder follows below mentioned steps for block based motion estimation[2]:

1. Calculate distortion measure of current block of current frame with a set of blocks in search region of previous frame.
2. Determine the lowest distorted block of previous frame and set it as matched block.
3. Determine the lowest distorted block of previous frame and set it as matched block.
4. Subtract the matched block with current block.
5. Encode and transmit the residue.

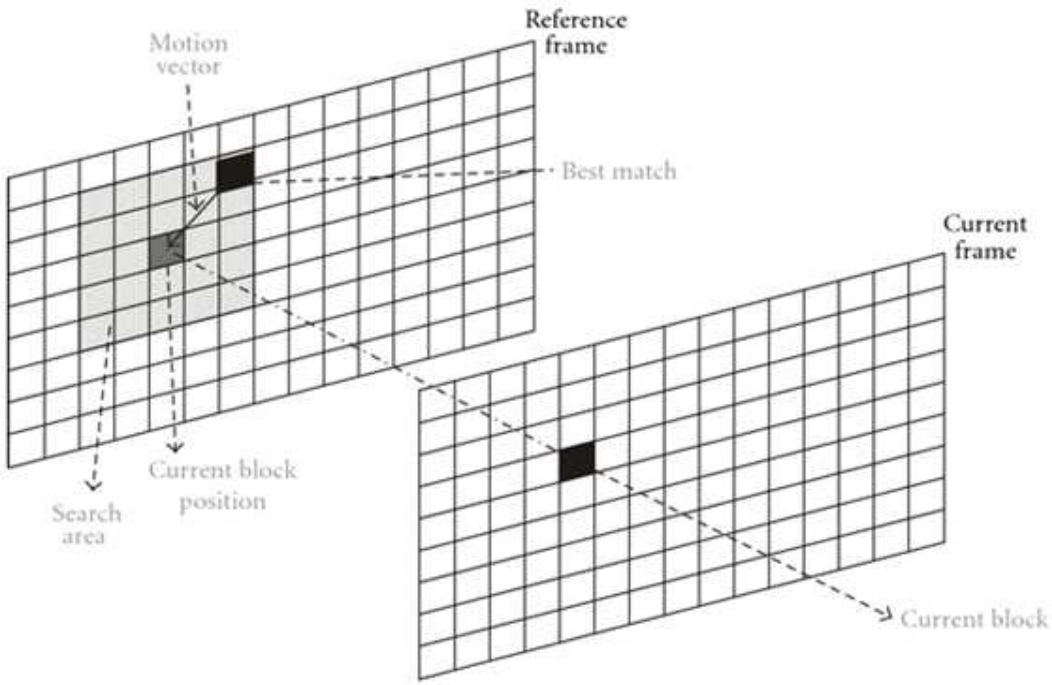


Figure 3.2: Block Matching Based Motion Compensation

6. Determine motion vector which is displacement with current block from previous block and encode and transmit it.

3.3 Popular Motion Estimation Algorithm in Video Coding

In order to accurate matching, theoretically searching for comparison need to be carried out every possible area in the reference frame. But this is impractical as it will add a huge computational complexity. In practical, a good number of searching is carried out within a defined a search window centred on the current block co-ordinate of previous frame. The size of the search window is very important and depends on several factors.

- **Resolution:** A larger window is better for higher resolution whereas respectively smaller window for lower resolution.

- **Motion:** A larger window is appropriate for high motion scene whereas small window for slow motion.
- **Computational complexity:** Higher the size of search window higher the computational complexity.

There are so many block matching based motion estimation algorithms available which estimate the motion balancing performance metric based on the requirements[3].

3.3.1 Exhaustive Search

This is commonly known as Full Search. It searches all possible blocks in the search window.So for –N/N search region.

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

It guarantees the true global minima in the search window but computational complexity is too high and so impractical for real time fast scenario.

3.3.2 Logarithmic Search (LS) Algorithm

Jain and Jain[4] introduced this search algorithm. This is a multi-stage search where step size is halved in each stage. In each step, it searches for four search points in '+' shape along with centre with S step size.

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

where n1= number of iterations where step size is unchanged, n2=number of times step size is halved.

This algorithm overcomes the huge computational overhead of full search but due to very small number of search points algorithm fails to reach true global minimum.

3.3.3 Three Step Search (TSS)

Koga et al.[5]proposed this algorithm is square search pattern with eight search points around the centre. This is a special case of N-step search and good approach over Logarithmic Search. . Here initial step size half of the maximum motion displacement.

Computational Complexity C.C. $= 1 + 8 \times (\log_2 (\mathrm{N} + 1))$

It gives better performance than logarithmic search and due to less complexity and optimum result this algorithm is adopted in many video coding standards.TSS has a good number of search points in the pattern. It does not have any early termination condition too. This makes the algorithm inefficient as it has to search a large search points though it is ready to go for termination. Apart from that is observed that it fails to detect small motion.

3.3.4 New Three Step Search (NTSS)

Reoxiang et al. [6] has proposed this algorithm which is an improved version of TSS. This is a center biased search algorithm with a half-way termination to reduce computational burden.

Computational Complexity C.C $\left\{ \begin{array}{ll} = & 17 \text{ for early termination} \\ = & 9 + 8 \times (\log_2 (\mathrm{N} + 1)) \text{otherwise} \end{array} \right\}$

Three step search method is good for stationary quasi stationary and slow motion and speed up the system with early termination. Thus this overcomes the problems of TSS but it is more complex than TSS too.

3.3.5 Four Step Search (FSS) Algorithm

Since NTSS adds intense computational overhead, FSS[7] is proposed. This is also centre biased based search method. Here initial step size one fourth of maximum displacement . It has for searching step with half way termination which is in second or third step.

Computational Complexity C.C. $= 9 + 5 \times \mathrm{n}_1 + 3 \times \mathrm{n}_2 + 8$

where n1 = number of iterations minima occurred at corners of 5×5 search pattern n2 = number of iterations minima occurred at face centres of 5×5 search pattern.

It reduces the computational complexity. In worst case the no. of search points of TSS, NTSS and FSS are 25, 33 and 27 respectively. This is more efficient in slow or quasi stationary motion.But this algorithm is susceptible to trap into of local minima.

3.3.6 Diamond Search (DS) Algorithm

Diamond Search[8] non rectangular center-biased motion estimation algorithm. This is first popular algorithm with another shape apart from rectangle and with no limitation of search steps. There are two types of pattern here one is Small Diamond Search Pattern (SDSP) and another is Large Diamond Search Pattern (LDSP) to take care slow and fast motion.

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

where n1 = no. of iterations the minimum SAD point occur at vertices of LDSP. n2 = no. of iterations the minimum SAD point occur at face centres of LDSP.

DS does not have fixed steps and so, it will search till it finds the minima according to two patterns. LDSP tries to explore the search region to find medium or fast motion and SDSP attempt to give a level of accuracy and also take care of quasi stationary case.This strategy is good to reduce the probability of local optimum trapping and thus able to converge to global minima.

3.3.7 Cross Diamond Search (CDS) Algorithm

DS searches 13 points to terminate for stationary block which is huge. In real time video sequences, 70%-80% micro blocks are stationary and thus DS is efficient here. To overcome this,Cross Diamond Search[9] is proposed which is Cross Shaped Pattern nine search points as shown

$$\text{Computational Complexity C.C} = \left\{ \begin{array}{c} 9 \ 1^{st} \text{ step termination} \\ 11 \ 2^{st} \text{ step termination} \\ 11 + 4 + 5 \times n_1 + 3 \times n_2 + 4 \quad \text{otherwise} \end{array} \right\}$$

where n1 and n2 are same as in DS.

This is improved version of Diamond search and computational complexity is lesser with DS keeping the quality intact.

3.3.8 Kite Cross Diamond Search (KCDS) Algorithm

KCDS[10] is also mini center-biased kite patterns based search technique and an improvement over CDS. In the first step,it uses small cross shaped pattern (SCSP).This is reduced further to decide stationary blocks[12]. This is the first popular algorithm which incorporates asymmetrical to boost up the search in motion area

$$\text{Computational Complexity C.C}=\left\{\begin{array}{c} 5\ 1^{st}\text{ step termination} \\ 9\ 2^{st}\text{ step termination} \\ 9+\mathrm{k}+5\times \mathrm{n}_1+3\times \mathrm{n}_2+4\text{otherwise} \end{array}\right\}$$

where n1 and n2 are same as in DS and k is the of new search points added during selection of first diamond.

3.3.9 Hexagonal Search (HEXS) Algorithm

All mentioned algorithms whether symmetric or asymmetric don't have omni directional search pattern.Ideally, omni directional shape is circle and here a circle-shaped search pattern is adopted to start in unbiased distribution of minimum search point based hexagon pattern[11]. This consists of two patterns one is large Hexagon pattern and another is small inner hexagon pattern.Large pattern tries to explore in all motion direction whereas small pattern gives the accuracy.

Computational Complexity=7+3× n+4

where n=no. of iterations the hexagon pattern is repeated. This algorithm is a fast algorithm with fewer search points and thus a significant improvement of DS.

3.3.10 Particle Swarm Optimization based Search (PSOB) Algorithm

All shape based motion estimation algorithm is some rule based and some fixed number of search point in the pattern. But above mentioned different search patterns are proposed to have less number of search points with a good accuracy but still it does not reach to the optimal level. Apart from that some of the algorithms suffer from poor accuracy as it traps to local minima. PSO based algorithm is a stochastic process which randomly

tried to find the global minima with swarm intelligence. Conventional PSO with random particle and velocity have huge computation complexity to converge to global minima. So, different researchers has adopted different techniques of particle initialization, particle history preservation and different stopping and controlling criteria to make a faster motion estimation techniques which is respectively accurate but has lesser number of search points.

3.4 Distortion Measure

Block Matching is performed by minimizing certain block distortion measures that provide the error energy.

Mean Square Error

Mean Square Error is defined as

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

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

Absolute Error (MAE)

MAE is calculated as

$$MAE = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \tag{3.2}$$

Sum of Absolute Differences (SAD)

Sum of Absolute Differences is calculated as

$$SAD = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |C_{ij} - R_{ij}| \tag{3.3}$$

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

3.5 Performance Parameter

There are few standard parameters measured to assess the performance of video compression. These are discussed as follows:

Compression Ratio (CR):

This is a measure to define the degree of compression achieved. It is defined as the ratio of size of original video to the compressed video or encoded video and given by

$$CR = \frac{Size\,of\,Original\,Video}{Size\,of\,Encoded\,Video} \tag{3.4}$$

Though compression is done by removing the redundancy, but usually at the expense of quality, a compression technique reduce entropy and thus bitstream required to generate. In general, greater the compression ratio, lesser the quality and thus system demands a trade-off between quality and compression.

Quality:

As mentioned, compression does the quality degradation so; it is highly required to maintain perpetual quality.

- **PSNR:** The widely used quality parameter used is peak signal to ratio (PSNR) and defined as

$$PSNR = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\frac{255^2}{(I_o(i,j) - I_r(i,j))^2}\right) \tag{3.5}$$

  where I_o(i, j) is the original frame and I_r(i, j) represents the reconstructed frame at the decoder.

  But PSNR is quantitative measure provides amount of distortions occurred in the process but do not consider human visual perpetual quality.

- **SSIM:** So, there is a new parameter structural similarity index measure (SSIM) used which is quantities measure to judge the visual quality as it compares of structural similarity, luminance and contrast[12]. This is defined as

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)\,(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{3.6}$$

where is μ mean, σ^2 is variance, $C_1 = (K_1L)^1$ and $C_1 = (K_1L)^2$ where L is the maximum value of the pixel (255 for 8-bit gray-scale images) and K1 and K2 are very small values close to zero(K<<1).

However these are still qualitative measure and fail to judge HVS quality and thus we require subjective comparison of different expert views.

Computational Complexity

Any video coding techniques require having less computational overhead and as discussed before motion estimation has the maximum (70% -80%) overhead. So, lesser the search point to find the best minima is lesser the computational complexity and faster the system.

Encoded Bitrate

This is the average number of bits encoded per second with a specified frame and defined as

$$Encoded\,Bit\,Rate = \frac{Size\,of\,Encoded\,video\,in\,Bits \times Frame\,Rate}{Total\,Number\,of\,Frames} \tag{3.7}$$

Rate Distortion (R-D)

Shannon's theorem fidelity and coding rate [39] has an important role in video compression research. Here,PSNR is plotted against the bitrate and PSNR generally increases as bitrate increases.Thus, R-D plot is to estimate the performance of a coding scheme and for comparing performances of difference coding schemes. The R-D lies entirely above is the better scheme.

Chapter 4

Proposed Motion Estimation Algorithm

4.1 Introduction

Different search algorithms proposed so far is to have less number of search points with a good accuracy but still it does not reach to the optimal level. Apart from that some of the algorithms suffer from poor accuracy as it traps to local minima. In block matching based motion estimation, sub image is checked to locate the best matched block in defined search window. So, motion estimation is considered as an optimization technique and all standard optimization technique can be used here to optimize the solution.Evolutionary algorithms (EA) are well known optimization algorithms based on stochastic nature of natural behavior of animals or biological genetics.These are very good at approximating solutions to different types of exiting problems because they are unaware of fitness function.Here author proposes some novel block matching based motion estimation based on two popular evolutionary technique, Particle Swarm Optimization (PSO) and Differential Evolutionary (DE).

4.2 Model Background

In the optimization field, PSO and DE are very popular because these algorithms.

- are very simple and straightforward to implement.
- perform better.
- have less number of control parameters.
- have lesser number space complexity.

4.2.1 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) being a heuristic algorithm (considered as an evolutionary algorithm[13]) is similar to movement of flock of birds aiming to find food. It is an optimization technique [14], with their initial locations and initial velocity may be chosen randomly or with some prior knowledge of problem. PSO makes almost no assumptions about the problem to be optimized and search in large solution space. Here the swarm is considered as particle, present in a search region. In each problem set, particles are considered as individual candidate solution. Initially, position and velocity is assigned to each particle randomly or some value based on the problem set. Each particle movement in search space is evaluated based on personal experience and neighboring particle experience. Personal best (pbest) is the best value achieved by individual particle. The best value among all personal best so far in the population, is known as global best, (gbest). In each iteration, position and corresponding velocity of each particle is updated based on its previous velocity, pbest and gbest ,(4.1)

$$v_i(t+1) = wv_i + c_1r_1(p_{i,best}(t) - x_i(t)) + c_2r_2(g_{best}(t) - x_i(t)) \tag{4.1}$$

The PSO algorithm consists of following module

Particle Initialization

$$X_i = \{x_{i1}, x_{i2}....., x_{in}\} \tag{4.2}$$

Velocity corresponding to each of the particles is defined as

$$V_i = \{v_{i1}, v_{i2}....., v_{in}\} \tag{4.3}$$

The positions of particles may be are initialized randomly in the range $[x_{min}, x_{max}]$ and corresponding velocities in the range $[v_{min}, v_{max}]$.

Algorithm 1: Particle Swarm Optimization

Input: Number of particles, nop;
Range of particles initial velocity $[v_{min}, v_{max}]$;
Maximum number of iterations, maxiter;
Cognitive learning rate c_1;
Social learning rate c_2;
Range of particles initial positions $[x_{min}, x_{max}]$;

```
while termination criteria not reached do
    Compute the fitness values of each particle;
    if k = 1 then
    pbest_value = fitness;
    pbest_position = pinitial_position;
    gbest_value = min (pbest_value);
    Set pbest_position equal to position corresponding to min (pbest_value)
    else   for i ← 1 to nop do
        if fitness (S_i) < pbest_value_i then
            pbest_value_i = fitness (S_i) ;
    if min(pbest_value) < gbest_value then
        gbest_value = min(pbest_value);
        Set gbest_value equal to the position corresponding to min(pbest_value)
    Update velocity and position of each particle in the population;
    Increment iteration count, k=k+1;
```

Particle pbest and gbest update

Fitness function is objective function need to be optimized. Now in each generation swarm finds best possible solution.This fitness function decides whether to update pbest

and gbest position or not.

Particle Position and Velocity Update

The updating of particle's position and velocity plays a key role in this optimization problem.Now in each generation the swarm intelligence accelerate the velocity of movement towards the best possible solution where there may exist the best possible solution. The velocity and thereby the position of the particles are updated.

$$v_i(t+1) = wv_i + c_1r_1(p_{i,best}(t) - x_i(t)) + c_2r_2(g_{best}(t) - x_i(t)) \tag{4.4}$$

The position of this particle is also an n-dimensional vector

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{4.5}$$

Where i is the index of the particle, i=1, 2,..., K; w the inertia weight; c1, c2 the positive acceleration constants referred to cognitive parameters r1 and social parameters; r2 uniformly distributed random numbers, within the interval [0,1]; t the number of iterations so far; pbest the position of personal best for the particle i; and gbest is the position of global best for the entire population

4.2.2 Differential Evolution (DE)

This is very popular evolutionary algorithm which can be considered as a advantageous combination of PSO and Genetic algorithm (GA). It gives very good results while compared to other optimization available in the literature. Compared to most competing PSO, DE

- has better performance
- variants is largely better than PSO variant over a wide variety of problems

But it requires sufficient number of chromosome for proper convergence to optimum result.

DE Basics

The current generation initial vector from which child or offspring has to take birth is called target vector or parent vector.The vector obtained from differential mutation operation is also known as donor vector or mutant vector. Finally an offspring generated by recombining the donor with target vector and is called as trial vector.After determining whether the target or trial can survive to the next generation, most fitting offspring take birth and treated as parent vector for the next generation.

Mutation

This is to add a sudden change or perturbation with a random element to change in the gene characteristics of a chromosome. In mutation, to generate each donor vector $\vec{V}_{i,G}$ corresponding to i^{th} target vector, other distinct target vectors e.g.$\vec{X}_{r_1{}^i,G}$,$\vec{X}_{r_2{}^i,G}$,$\vec{X}_{r_3{}^i,G}$, $\vec{X}_{r_4{}^i,G}$and $\vec{X}_{r_5{}^i,G}$ are sampled randomly where $r_1{}^i$, $r_2{}^i$, $r_3{}^i$, $r_4{}^i$ and $r_5{}^i$ are mutually exclusive in the current population.$\vec{X}_{best,G}$ is commonly known parent best of the current generation. Based on the difference vector of randomly sampled target vector, DE family by Stron and Price has following well-known mutation schemes

DE/rand/1:

$$\vec{V}_{i,G} = \vec{X}_{r_1{}^i,G} + F.(\vec{X}_{r_2{}^i,G} - \vec{X}_{r_3{}^i,G}) \tag{4.6}$$

DE/best/1:

$$\vec{V}_{i,G} = \vec{X}_{best,G} + F.(\vec{X}_{r_1{}^i,G} - \vec{X}_{r_2{}^i,G}) \tag{4.7}$$

DE/target-to-best/1:

$$\vec{V}_{i,G} = \vec{X}_{i,G} + F.(\vec{X}_{best,G} - \vec{X}_{i,G}) + F.(\vec{X}_{r^i{}_1,G} - \vec{X}_{r^i{}_2,G}) \tag{4.8}$$

DE/best/2:

$$\vec{V}_{i,G} = \vec{X}_{r^i{}_1,G} + F.(\vec{X}_{r^i{}_2,G} - \vec{X}_{r^i{}_3,G}) + F.(\vec{X}_{r^i{}_4,G} - \vec{X}_{r^i{}_5,G}) \tag{4.9}$$

DE/rand/2:

$$\vec{V}_{i,G} = \vec{X}_{best,G} + F.(\vec{X}_{r^i{}_1,G} - \vec{X}_{r^i{}_2,G}) + F.(\vec{X}_{r^i{}_3,G} - \vec{X}_{r^i{}_4,G}) \tag{4.10}$$

Here F is known as scaling factor and an important sensitive controlling parameter as it decides what percent of difference vector is to be added in mutation.

Algorithm 2: DIFFERENTIAL Evolution

Read the values of popsize, maxgen from user.;
Set the generation count i=0;
Initialize the popsize of Chromosomes $\mathrm{P^i} = \left\{\mathrm{p^i_1}, \mathrm{p^i_2}, \ldots, \mathrm{p^i_{popsize}}\right\}$ with $\mathrm{p^i_j} = \left\{\mathrm{p^i_{j,1}}, \mathrm{p^i_{j,2}}, \ldots, \mathrm{p^i_{j,K}}\right\}$ for j=1,2,3.....popsize, where K= dimension of each chromosome, $\mathrm{p^i_{j,K}}$ is the kth allele/gene of the jth chromosome in i^{th} iteration having dimension D.;
for $j \leftarrow 1$ *to popsize* **do**
    Calculate the fitness value of chromosome i.e. fitness x^0_{j} ;
while *termination criteria is not satisfied* **do**
    for $j \leftarrow 1$ *to popsize* **do**
        // Perform Mutation $v^i_j = Mutation(x^i_j, P^i)$; with $\mathrm{m^i_j} = \mathrm{Mutation}\left(\mathrm{p^i_j}, \mathrm{P^i}\right)$ is the mutant vector corresponding to j^{th} chromosome in i^{th} iteration, where $v_(j,k)^i$ is k^{th} element of the mutant vector having dimension D;
        // Perform Crossover
        $\mathrm{t^i_j} = \mathrm{Crossover}\left(\mathrm{p^i_j}, \mathrm{m^i_j}\right)$; with $\mathrm{m^i_j} = \left\{\mathrm{m^i_{j,1}}, \mathrm{m^i_{j,2}}, \ldots, \mathrm{m^i_{j,K}}\right\}$ is the trial vector corresponding to j^{th} chromosome in i^{th} iteration, where $u_(j,k)^i$ is k^{th} element of the trial vector having dimension D;
        // Perform Local optimization
    for $j \leftarrow 1$ *to popsize* **do**
        Calculate the fitness value of trial vector i.e. fitness
    // Perform Selection **for** $j \leftarrow 1$ *to popsize* **do**
        $\mathrm{p^{i+1}_j} = \mathrm{Selection}\left(\mathrm{p^i_j}, \mathrm{t^i_j}\right)$;
    Increase the generation count i=i+1;

Crossover

To diversify the characteristic, donor vector after mutation, exchanges its characteristics with the target vector through crossover operation to generate the trial vector$U_{i,G}$. The

popular crossover is binomial crossover which is as follows

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & if(rand_{i,j}[0,1] \leq Cr \ or \ j = j_{rand}) \\ x_{j,i,G} & otherwise \end{cases} \tag{4.11}$$

Here Cr is known as crossover rate and very sensitive controlling parameter in DE,$rand_{i,j}[0,1]$ is random number with uniform distribution and $j_{rand} \in [1,2,3,.....D]$ D is the dimension of the vector.

$$Cr^i = cauchyrnd(mean, 0.1) \tag{4.12}$$

Selection

This is mainly to determine whether the target or trial can survive to the next generation.

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & if \ \ f(\vec{U}_{i,G} \leq f(\vec{X}_{i,G})) \\ \vec{X}_{i,G} & if \ \ f(\vec{U}_{i,G} \geq f(\vec{X}_{i,G})) \end{cases} \tag{4.13}$$

Parameter Adaptation

The mutation scale factor F and crossover constant CR is very sensitive for correct and faster convergence. A large number research is done for the proper value of F and CR but different test field and domain if problem require different value. So self-adaption is highly required for these sensitive parameters. Many self-adaptive schemes are proposed and among them commonly used are generating of parameter with Cauchy Distribution or Normal Distribution.

$$Cr^i = cauchyrnd(mean, var) \tag{4.14}$$

$$F^i = normrnd(mean, var) \tag{4.15}$$

For both the cases, initial mean is updated in each generation and accordingly Cr and F is updated by the following equation.Where α is learning rate and lies in [0 1] and

$$mean = \alpha \times mean + (1 - \alpha) \times meann \tag{4.16}$$

Here $meann$ is the average value corresponding parameter successful passed in the selection. For CR,

$$meann = avg(Cr_{success}) \tag{4.17}$$

For F

$$meann = avg(F_{success}) \tag{4.18}$$

4.3 Motion Analysis

In the real life video sequece it is observed that

- 70 % - 80% of Macro – Blocks are static or quasi static.
- Few of rest have the tendency to follow the same block from previous frame
- Few of the Macro-Blocks have the motion with the influence of nearby spatial MBs

Based on this analysis,particle or chromosome are placed in solution space for faster and accurate convergence for the following proposed algorithm.

4.4 Proposed Algorithms

4.4.1 Hybrid PSO based (HPSOB) Motion Estimation

Initial Particle Position

In the conventional PSO, the initial particle position is randomly chosen so particle gets their initial position in unbiased fashion in the solution space. But if initial position of particles is near to global best then, the particles converge very fast. This can only be done if the system has knowledge of solution neighborhood has some prior knowledge.Here some strategy is to be followed for faster and accurate convergence.

Strategies for first prediction frame in GOP

Strategy 1: One particle is to be placed at [0, 0].

Basis: As per the motion analysis in motion estimation, 70% - 80% block has the converged output at [0, 0] or around it due to Stationarity or quasi Stationarity.

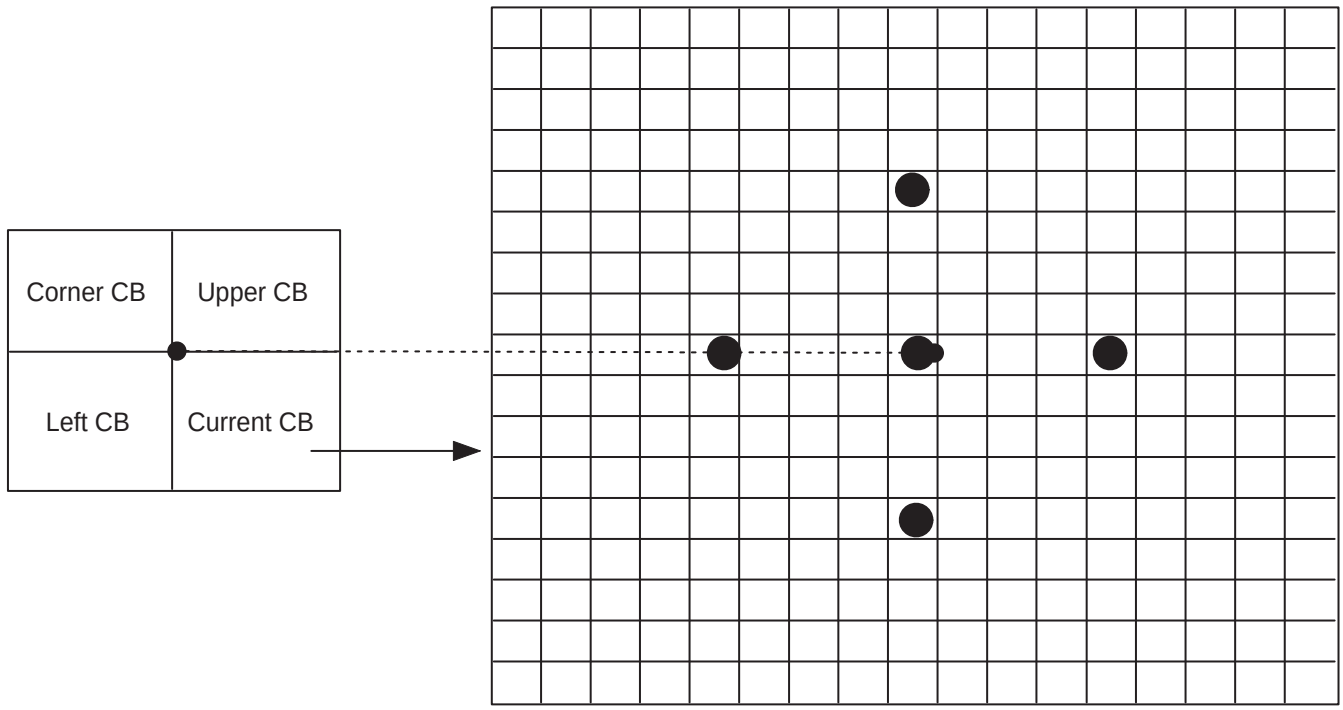


Figure 4.1: Particle Initialization for first P frame of GOP of HPSOB Motion Estimation

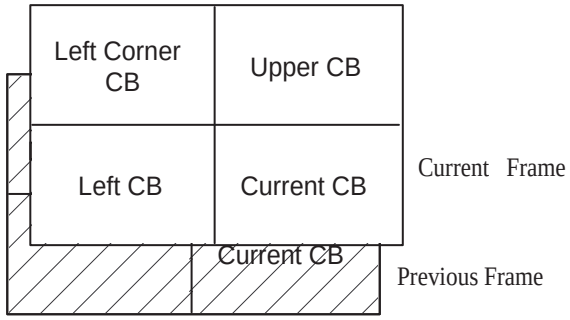


Figure 4.2: Initial Particle position rest P frames of GOP of HPSOB Motion Estimation

Strategy 2: Place two particles, one at left adjacent spatial block motion vector distance from [0, 0] and another at upper adjacent spatial block motion vector distance from [0, 0]

Basis: The motion in real video sequence is smooth and strongly correlated with spatially adjacent block.

Strategy 3: Place four more particles in ±sz search window at a distance sz/2 from [0, 0] which are [sz/2,0],[0,sz/2],[-sz/2,0],[0,-sz/2].

Basis: To allow the particle to explore the other area , these four particles are positioned. As the search is ±sz, the fast motion can have up-to a displacement of ±sz. To keep in mind slow, medium and faster motion particle locations are at medium value of highest possible displacement which is sz/2. The particles are at vertical or horizontal direction taking into consideration that vertical and horizontal motion are more than diagonal motion and directional displacement can be taken care by the vertical and horizontal particles while exploring.

Strategies for other prediction frame in GOP

Here the particle position will be exclusively on spatial and temporal correlation basis. As the previous frame motion vector is determined already, current block has prior knowledge of the motion vector of previous frame. So, here four particles are to be placed, among them three are based strategy 1 and strategy 2 and strategy 4 where strategy 4 is as follows.

Strategy 4: Place one particle;at a distance of previous frame same block motion vector value

Basis: Video frame and block in frame is highly temporally correlated and

Further refinement is done based on discarding the same positioned particle.

Controlled Criteria

In conventional PSO, if the number of iterations increases, probability to reach actual global minima is higher but it will add a huge computational overhead. But here goal is to minimize the computational overhead. Again this will not allow the algorithm to get a refined result.So, the termination criteria should control the iteration adaptively, to maintain accuracy whereas lesser computational complexity. The proposed algorithm maintains two level of control. It allows the particle to roam around in the solution space up-to I_{end} where it does not go for termination but only skips calculating fitness function if it follows the same from the previous generation. Now, the particles are freely explored the solution space and ready for convergence. So, iteration after I_{end} , some checking are there to control the movement to go for termination. To get an optimized output at a faster rate we adopted some controllers in the algorithm. They are as follows.

Criterion 1: In each iteration, skip calculating fitness function if it follows the same from the previous generation.

Basis: This will restrict to calculate fitness function unnecessarily as it is already in the same location.

Criterion 2: After I_{end} in each iteration, if any of the particle position is same as previous, then go for termination.

Basis: Up-to I_{end} particle get sufficient time to explore and converge to the best result and if a particle still stays in the same location for consecutive iteration after exploring so much then it can be concluded that this particle reach to optimal best value and fine refinement is not possible or if possible then require high computational cost.

Criterion 3: After I_{end} , in each iteration if any of the particle position is at [0, 0], then go for termination.

Basis: This condition is an assumption of stationarity. After exploring the solution space up-to I_{end} if particle resides at [0 0], it signifies that the particle finds the best value here which pointing out the block to be stationary.

Criterion 4: In any case in each iteration, if best fitness value reaches to particular threshold, go for termination.

Basis: The threshold value is sufficient to get good compression. Further refined value may be possible in expense of higher computational burden which is undesirable.

Criterion 5: In any case, if updated velocity crosses some defined range then set the velocity accordingly.

Basis: This is restricting the velocity of the particle so that it don't diverse and come out from solution space.

The stopping criteria are based on the idea that the algorithm already has explored so many places in the solution space to find best value or nearly. If it fails to find till now, the stopping criteria shows the algorithm should go for termination rather than iteration as it already achieved or further iteration don't give much fine refinement.

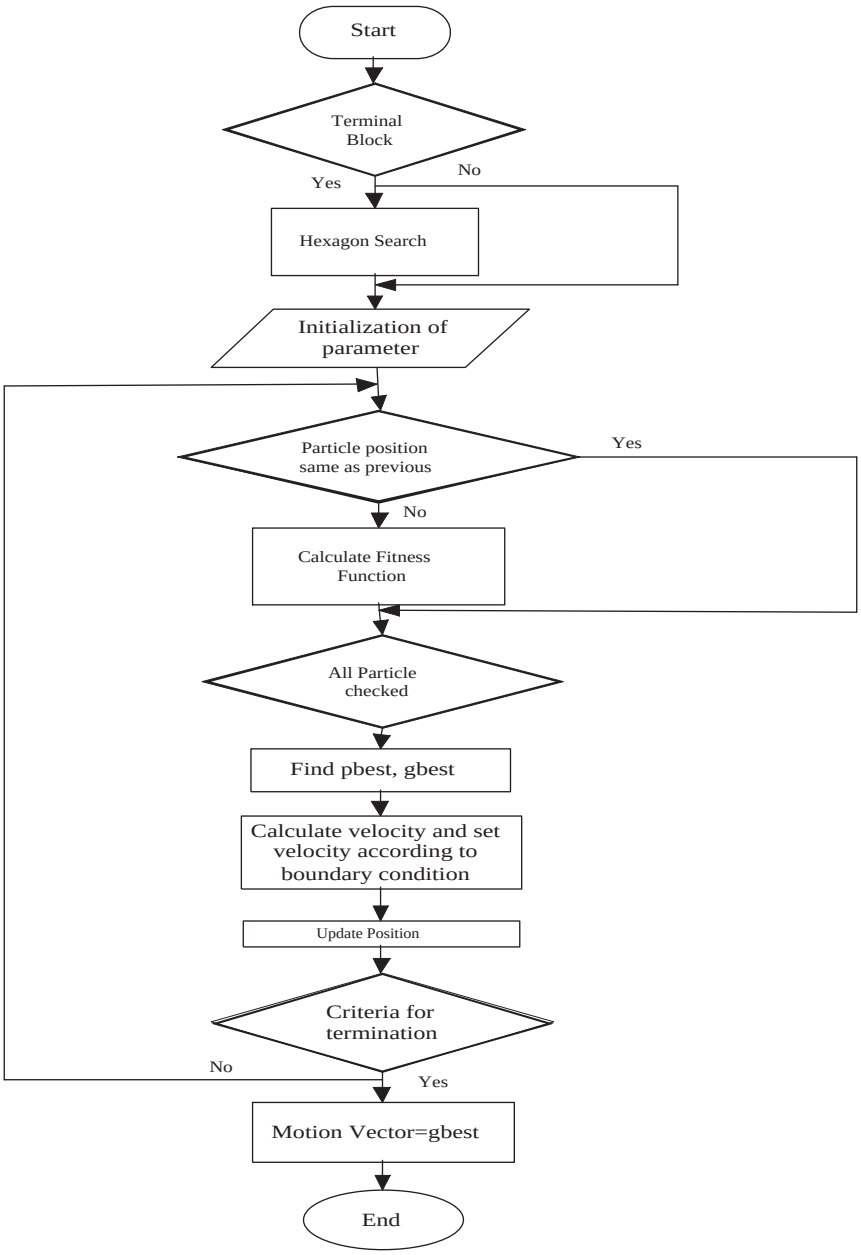


Figure 4.3: Flow Chart of Hybrid PSO based Motion Estimation

Boundary block processing

All terminal blocks e.g. first and last row and column blocks for first prediction frame of 'IPPP' and the first block of rest prediction frames of GOP is processed by hexagonal searching method instead of particle swarm optimization techniques.In boundary blocks, spatially adjacent block according particle initialization rule is less or missing so

PSO based algorithm takes much time to converge or may give less accurate value in lesser iteration.Among the all standard motion estimation algorithm, hexagonal search is adopted here as it is faster and checks almost all possible direction and good at slow and fast motion too.For other block is to be processed by following proposed algorithm.

Algorithm

The algorithm is shown in the figure 4.3 .First it checks whether the processed block is boundary blocks or not. If it is a boundary block it process according to boundary block otherwise it process according to designed PSO based algorithm for first prediction frame. First the parameters associated with PSO algorithm discussed earlier in this chapter are initialized. Particles are initialized as per the particle initialization rule discussed already. After that it just follow conventional PSO algorithm whose pseudo code discussed already. Each time, particle checks the controlled criteria and accordingly take the decision of skipping or termination.

4.4.2 Stationary Aware Hybrid PSO based (SAPSOB) Motion Estimation

Initial ParticlePosition

particle initialization is same as in previous frame.

Controlled Criteria:

All the controlled criterion Criterion 1 to Criterion 5 are applicable here. Here another two more criteria are adopted which is discussed below.

Criterion 6: Before start of iteration. if at [0, 0] $Fitness < Fitness_{thres}$, go for termination.

Basis : This is to eliminate the unnecessary processing to the stationary blocks or the blocks reached to the value where no more refinement is required.

Criterion 7: After a particular generation I_{endk}, if two of best among pbest are same then go for termination.

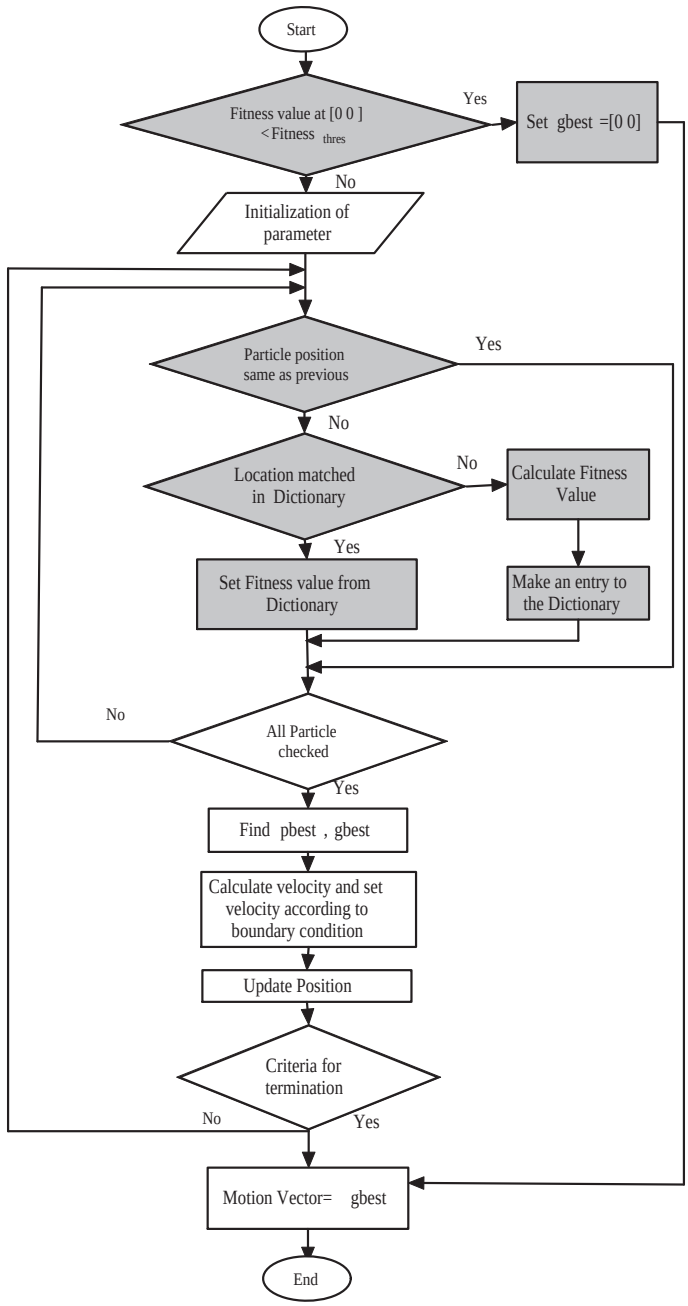


Figure 4.4: Flow Chart of SAPSOB Motion Estimation

Basis : Up-to I_{endk} the particles explored the solution space very well and ready to converge. In this scenario, if two of best among pbest are same it signifies that possibly other particle will soon converge to here.

Boundary block processing

All terminal blocks e.g. first and last row and column blocks for first prediction frame of 'IPPP' and in consecutive prediction frames only the first block is processed by Kite Cross Diamond Search (KCDS) searching method instead of particle swarm optimization techniques.In boundary blocks, spatially adjacent block according particle initialization rule is less or missing so PSO based algorithm takes much time to converge or may give less accurate value in lesser iteration.Among the entire standard motion estimation algorithm, KCDS search is adopted here as it is faster and checks almost all possible direction and good at slow and fast motion too.For other prediction frame in GOP, the first block is to be processed by KCDS search and rest by proposed method.

Algorithm

The detailed flow chart of proposed algorithm is shown in figure 4.4.It first checks whether the processed block is boundary blocks or not. If it is a boundary block it process according to Hexagonal Search otherwise it process according to designed PSO based algorithm. First the parameters associated with PSO algorithm discussed earlier in this chapter are initialized. Particles are initialised as per the particle initialization rule discussed already. After that it first check whether the block is stationary or not and if stationary then go for termination otherwise it just follow conventional PSO algorithm. Each time, particle checks the controlled criteria and accordingly take the decision of skipping or termination.Here location history of particles is preserved in a dictionary. Each time whenever particle is a new location, it first searches in dictionary. If any particle in the population already traversed the location, there should be an entry to the dictionary and particle will fetch the value from dictionary otherwise it calculates the fitness and makes a new entry to the dictionary.

4.4.3 Motion Aware DE and PSO based (MADEPSOB) Motion Estimation

In this proposed algorithm the block searching is to be done in two separate ways. In 'IPPP' format, the first prediction frame processing will be based on differential evolution method and rest of the prediction frame on particle swarm optimization techniques based SAPSOB algorithm, discussed already.

Initial Particle Position

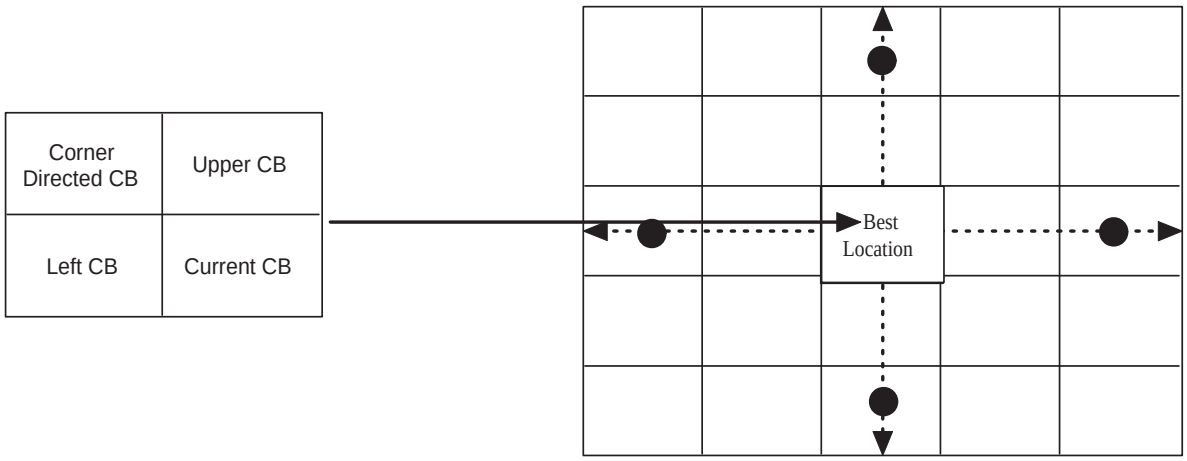


Figure 4.5: Particle Position of MADEPSOB algorithm

For first prediction frame the chromosomes in DE algorithm are to be positioned in two ways. At the start of the algorithm, four chromosomes are to be positioned in the search region.

Step 1: Position the chromosomes according to strategy 1 and strategy 2 discussed already.

Step 2: This is based the following strategy.

Strategy 7: Find the best chromosomes among four and treat as parent best.Here block motion is segregated into four parts apart from stationary and thus chromosomes are positioned initially

Quasi-Stationary: SAD>=a and <b

Position chromosome at spatially co-related position including [0,0]

Slow Motion: SAD>=b and <c

Position chromosome at sz/8 away from parent best

Medium Motion: SAD>=c and <d

Position chromosome at sz/4 away from parent best

Fast Motion: SAD>=d and <e

Position chromosome at (sz/4+sz/8) away from parent best.sz is half of the size of the block.

Basis : Video frame and block in frame is highly temporally correlated .The best fitted parent among all above mentioned chromosome is evaluated and treated as parent best. So, actual best should reside a neighborhood any of the five chromosomes. This spanned neighborhood region is to be decided by the type of motion. It is obvious that if it is slow motion, the actual best will be nearer to the parent best, or if it is fast motion it will be respectively far away. As per this analysis, the chromosomes of the second step processing are positioned according the motion of block. As mentioned already, the above is for DE based motion estimation for the first prediction frame of 'IPPP' and the rest of the frame will processed by SAPSOB algorithm and the initial particle position is as follows.

Controlled Criteria:

This is same as per SAPSOB motion estimation algorithm discussed before.

Terminal Block Processing

The first block is to be processed by kite cross diamond search for all prediction frame in GOP.

Algorithm

Here algorithm first checks whether block is boundary block or not and if it is boundary block it process by KCDS algorithm otherwise it checks the stationrity condition and if it matches then go for termination otherwise process according to proposed DE based motion estimation algorithm. The parameters of DE algorithm is initialized first. The

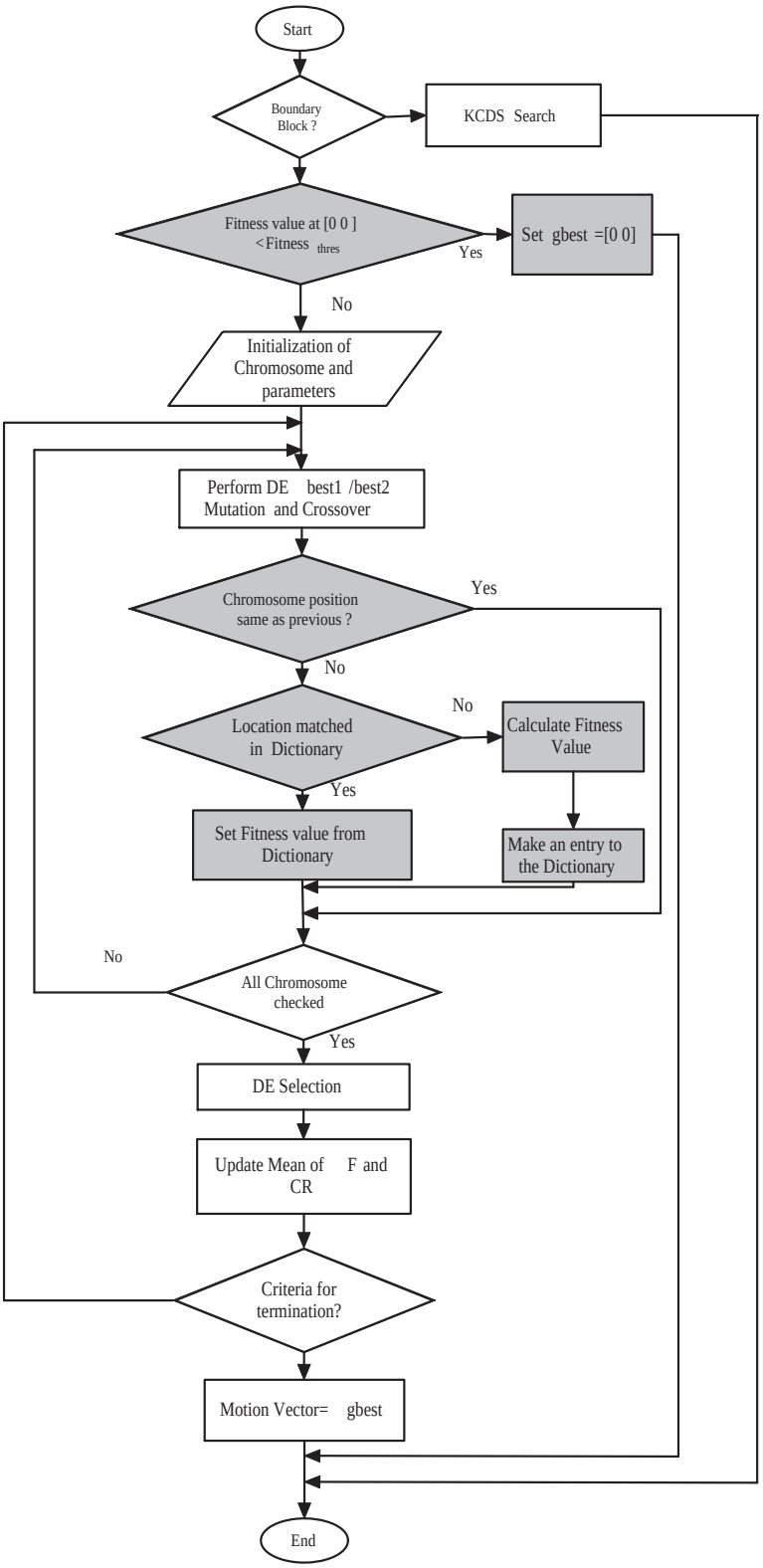


Figure 4.6: Flow Chart of Motion Aware DE and PSO based Motion Estimation

chromosomes are initialized according to chromosome initialization rule mentioned. In DE algorithm. The first block of operation in DE algorithm is mutation and there are so many mutation schemes available among which DE best1 or DE best2 is adopted here. These two schemes are very efficient in small population as it produce mutant with a difference vector scaled value to parent best. The CR and F parameters of DE is self updated. We defined the following self adaptive schemes for our algorithm.CR will be generated with Cauchy Distribution with initial mean 0.5 and variance 0.3 and F by Normal distribution with initial mean 0.5 and variance 0.3. The flow chart of the algorithm is shown in the figure 4.6.

$$Cr^i = cauchyrnd(mean, 0.1) \tag{4.19}$$

$$F^i = normrnd(mean, 0.3) \tag{4.20}$$

where mean is updated according to the equation.

4.4.4 Motion Directed PSO based Motion Estimation (MDPSOB)

Initial Particle Position:

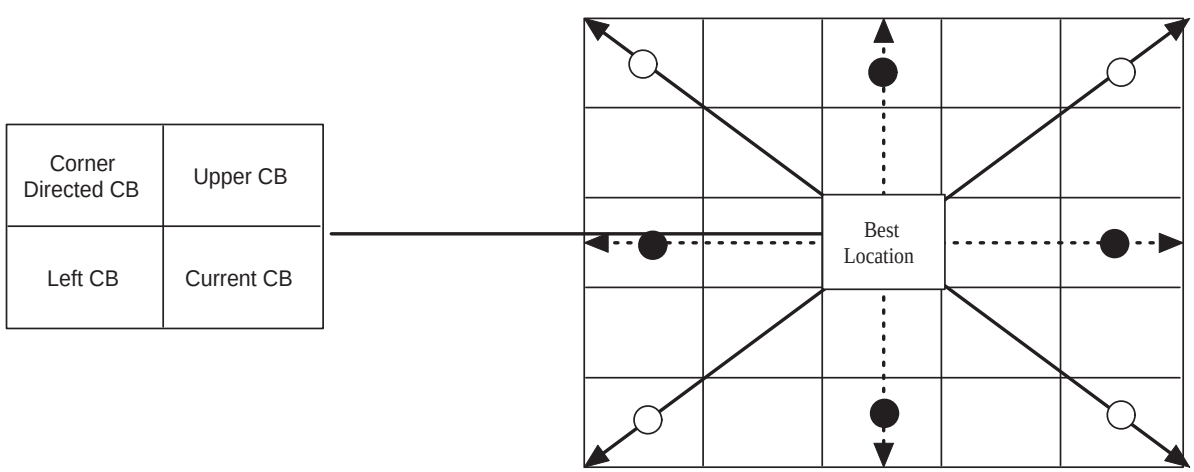


Figure 4.7: MDPSO based Motion Estimation

For first prediction frame the particles in DE algorithm are to be positioned in two ways. At the start of the algorithm, four particles are to be positioned in the search region.

Step 1: Position the particle according to strategy 1 and strategy 2 discussed already.
Step 2: This is based on the following strategies. Position the particle according to strategy 1 and strategy 2 discussed already.

Strategy 8: Find the best particles among four and treat as particle best. Find the angle of particle best from the origin. Let the angle is β . Then Here block motion is segregated into three parts apart from stationary and thus particles are positioned initially

Slow Motion: If SAD$>= a$ and $< b$

Position particles at spatially co-related position including [0,0]

Medium Motion: If SAD$>= b$ and $< c$

Position first particle X1 as follows

$X1\,(1,1) = particlebest\,(1) + (sz/4) \times round\left(cos\,(\beta)\,\sqrt{2}\right)$

$X1\,(1,2) = particlebest\,(2) + (sz/4) \times round\left(\sin\,(\beta)\,\sqrt{2}\right)$

The other three particles X2,X3,X4 are as follows

$X2(1,1) = X1(1,1) \times \cos\theta - X1(1,2) \times \sin\theta$

$X2(1,2) = X1(1,1) \times \sin\theta + X1(1,2) \times \cos\theta$

$X3(1,1) = X2(1,1) \times \cos\theta - X2(1,2) \times \sin\theta$

$X3(1,2) = X2(1,1) \times \sin\theta + X2(1,2) \times \cos\theta$

$X4(1,1) = X3(1,1) \times \cos\theta - X3(1,2) \times \sin\theta$

$X4(1,2) = X3(1,1) \times \sin\theta + X3(1,2) \times \cos\theta$

Fast Motion: If SAD$>= c$

Position first particle X1 as follows

$X1\,(1,1) = particlebest\,(1) + (sz/2) \times round\left(cos\,(\beta)\,\sqrt{2}\right)$

$X1\,(1,2) = particlebest\,(2) + (sz/2) \times round\left(\sin\,(\beta)\,\sqrt{2}\right)$ The other three particles X2,X3,X4 are as follows

$X2(1,1) = X1(1,1) \times \cos\theta - X1(1,2) \times \sin\theta$

$X2(1,2) = X1(1,1) \times \sin\theta + X1(1,2) \times \cos\theta$

$$X3(1,1) = X2(1,1) \times \cos\theta - X2(1,2) \times \sin\theta$$
$$X3(1,2) = X2(1,1) \times \sin\theta + X2(1,2) \times \cos\theta$$
$$X4(1,1) = X3(1,1) \times \cos\theta - X3(1,2) \times \sin\theta$$
$$X4(1,2) = X3(1,1) \times \sin\theta + X3(1,2) \times \cos\theta$$

Basis: The best fitted particle among all above mentioned particles is evaluated and treated as particle best. So, actual best should reside a neighborhood particle best value. This spanned neighborhood region is to be decided by the type of motion and direction of the motion. There is a high possibility that current best will follow the particle best so for accurate convergence particle should be in that cloud direction wise. Here according the equation given we segregated the motion direction into 8 parts shown in the figure, $0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°$,$360°$ The best particle may have these directional options only. Now one particle is to be positioned sz/4 distance along the direction assuming the block direction is same but as the motion is medium it can move some path along the direction. For sufficient exploration or direction change, three more particles is set which is same sz/4 away from the particle best but$90°$ away from each other. For fast motion, obviously particle is set It is obvious that if it is slow motion, the actual best will be nearer to the parent best, or if it is fast motion it will be respectively far away. As per this analysis, the chromosomes of the second step processing are positioned according the motion of block. As mentioned already, the above is for DE based motion estimation for the first prediction frame of 'IPPP' and the rest of the frame will processed by SAPSOB algorithm and the initial particle position is as mentioned.

4.5 Simulation and Result

To demonstrate the performance of the proposed motion estimation algorithm with well-known methods like Full Search (FS), Logarithmic Search (LS), Three Step Search (TSS), Four Step Search (4SS) , Diamond Search (DS), Hexagon Search (HS), Cross Diamond Search(CDS), Kite Cross Diamond Search (KCDS) and PSO based search [15] with different standard test cases used shown in the table 4.2 which includes three standard video formats discussed in the table 4.1

Table 4.1: Video Sequence Formats

Format	**Resolution**	Application
CIF	352×288	Video conferencing, over ISDN/Internet, H261,H263 ,CCTV, PAL
QCIF	176×144	Small device, CCTV, Video Telephony over wiered/wireless , H.263
SIF	352×240	VCD, Web, Offline Editing ,NTSC

Table 4.2: Test Video Sequence

Sequence	**Characteristics**
Stefan	Very complex, Contains zooming, panning both object and camera motion, fast change in object tracked and background
Football	Multiple object tracking, camera and object motion, panning, fast, boundary line motion, complex
Soccer	Multiple object, multi-directional movement, slow,medium and fast motion, camera motion, complex
Coastguard	High movement in boundary, camera is moving with motion object, background also changing
Bowing	Static background , slow and fast motion along with boundary motion, no camera motion
Tennis	Medium, fast and abrupt
Husky	High camera motion, zooming panning , tracking, scene change

. The inputs are considered as 4:2:0 YUV format. SSIM are measured to find the quality of the output with K1=0.01 and K2=0.03. Encoded bit rate is measured at frame.The fitness function is SAD. sz is taken as 8.Encoded bitrate is measured in different quantization value (Q value) as 24, 28, 32 and 36 and plotted against PSNR to analyze the rate distortion curve. Average search points are average number of searching for block matching done for each input sequences. The block size is 16 x16 and searching window is$-8/+8$.The test video signal is chosen such that we can test slow motion, wide variety of neighboring block, zooming,panning and other complex scenarios.

For all proposed PSO based algorithm, c1=.25 and c2=.5 for greedy search. For HP-SOB, for first prediction frame, maximum number of iteration is set as 8 and I_{end}=2.The velocity is restricted to [-6,6] but location update is not bounded Threshold ot cut off SAD value $Fitness_{thres}$=175.For the rest of P frames in the GOP, maximum iteration I_{max} = 5 , I_{end}=3 and $Fitness_{thres}$=250.The maximum iteration is less as the positioned particle has spatial and temporal knowledge but I_{end}=3 set to explore more due to lesser number of particles. For SAPSOB first P frame,I_{max} = 8 and I_{end}=3 and velocity is restricted in [-7, 7]. For other P frame in GOP,I_{max} = 6 and I_{end}=4. Thus, it allows the particle to explore more before going to first termination criteria checking. For MADPSO first P frame in GOP, $Fitness_{thres}$=250, intial paticle position range value a=250, b=450,c=700, d=1100,d=1500. Initial mean of CR and F is 0.5.I_{max} = 6 and I_{end}=3. Search is restricted in Search region of -8/+8.For other P frame it follows the SAPSOB algorithm.For MDAP-SOB, the parameter is same apart from the intial paticle position range value which is a=250, b=450,c=1100 .

All the simulation is carried out in MATLAB version 7.0.10.499 (R2010a) on an Intel® core(TM) i5-2400 CPU @ 3.10 GHz and the results are tabulated in the following pages

Table 4.3: Parameter Comparison For Test Sequence: Tennis.cif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	34.9713	0.82714	11.36082	194.7091	1622.418
LS	34.94085	0.826263	10.76587	18.0377	1712.077
TSS	34.89098	0.824621	10.75214	19.15227	1714.263
NTSS	34.88015	0.824139	10.35146	20.15252	1780.618
4SS	34.94281	0.826287	10.56153	17.56288	1745.202
DS	34.93295	0.825697	11.21498	17.79124	1643.516
HS	34.91619	0.825148	10.99164	12.46048	1676.912
CDS	34.96212	0.826459	10.95599	16.991	1682.368
KCDS	34.95455	0.826504	11.13518	14.50779	1655.294
PSOB	34.95663	0.826537	11.22128	14.13124	1642.593
HPSOB	34.94536	0.82582	10.93538	6.840576	1685.538
SAHPSOB	34.9458	0.826138	11.224	8.923333	1642.196
MADEPSOB	34.95031	0.826282	11.24586	10.54367	1639.004
MDPSOB	34.94606	0.826278	11.26251	10.03758	1636.58

Table 4.4: Parameter Comparison For Test Sequence: Stefan.cif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitrate(kbps)
FS	34.6166	0.93150	7.8928	194.7090	2335.272
LS	34.5643	0.93035	7.7711	17.634	2371.836
TSS	34.5369	0.929281	7.6204	19.1226	2418.768
NTSS	34.5264	0.92888	7.36063	21.81209	2504.131
4SS	34.5578	0.93023	7.5821	17.920	2430.957
DS	34.5638	0.93010	7.5972	19.8395	2426.128
HS	34.5245	0.92872	7.1379	12.0396	2582.240
CDS	34.5721	0.928721	7.1379	12.0396	2582.240
KCDS	34.5788	0.93037	7.5043	16.123	2456.178
PSOB	34.5960	0.93099	7.8717	16.6218	2341.5462
HPSOB	34.5760	0.930282	8.0963	13.6057	2276.5881
SAPSOB	34.5856	0.93084	8.1909	10.30163	2250.3012
MADEPSOB	34.5859	0.93103	8.15000	12.2398	2261.5944
MDPSOB	34.58580	0.93117	8.2252	11.8451	2240.9001

Table 4.5: Parameter Comparison For Test Sequence: Football.cif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	36.6029	0.87643	11.8169	196.878	1871.758
LS	36.3647	0.86831	11.5177	20.8216	1920.374
TSS	36.2337	0.86392	11.3392	19.3412	1950.618
NTSS	36.2312	0.86380	11.1515	23.2212	1983.429
4SS	36.3753	0.86862	11.3747	21.7647	1944.510
DS	36.3860	0.86897	11.4902	29.8613	1924.979
HS	36.3715	0.86825	11.3160	19.6281	1954.603
CDS	36.4151	0.86991	11.4130	30.4228	1937.993
KCDS	36.4790	0.87185	11.4657	27.6467	1929.079
PSOB	36.5752	0.87574	11.7028	21.5822	1890.0012
HPSOB	36.3988	0.86992	11.6960	15.4724	1891.0923
SAPSOB	36.4717	0.87294	12.3530	17.5443	1790.5263
MADEPSOB	36.4932	0.87379	12.3708	18.2769	1787.9445
MDPSOB	36.4762	0.87327	12.4558	19.8230	1775.7501

Table 4.6: Parameter Comparison For Test Sequence: Soccer.cif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	35.6913	0.84483	15.5326	196.8787	1423.994
LS	35.647	0.84352	15.02147	18.7151	1472.4519
TSS	35.6103	0.84242	14.91315	19.3378	1483.1466
NTSS	35.5645	0.84071	14.19343	21.6675	1558.3548
4SS	35.6115	0.84232	14.4590	18.3906	1529.7309
DS	35.6507	0.84348	15.12464	20.8082	1462.4076
HS	35.5417	0.84021	14.0192	13.8619	1577.7183
CDS	35.6026	0.8416	14.3317	23.2588	1543.314
KCDS	35.6484	0.8432	14.8718	17.8281	1487.269
PSOB	35.6940	0.84485	15.4671	15.1091	1430.022
HPSOB	35.6667	0.84393	15.2253	10.9751	1452.7308
SAPSOB	35.6803	0.84452	15.4652	9.5294	1430.1993
MADEPSOB	35.6766	0.84441	15.3628	10.3392	1439.7285
MDPSOB	35.6860	0.84463	15.4488	9.9517	1431.7173

Table 4.7: Parameter Comparison For Test Sequence: Coastguard.cif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	34.7278	0.875432	11.67287	196.8787	1894.8549
LS	34.71832	0.875117	11.49486	17.6008	1924.1976
TSS	34.70685	0.874689	11.49042	19.2109	1924.9413
NTSS	34.64026	0.87239	10.67478	16.0975	2072.0223
4SS	34.6654	0.873265	10.89222	15.20194	2030.6595
DS	34.7264	0.87532	11.60884	16.70015	1905.306
HS	34.67202	0.873548	11.00433	10.12454	2009.9724
CDS	34.73461	0.875537	11.56145	16.87022	1913.1153
KCDS	34.72996	0.875406	11.57836	13.8683	1910.3211
PSOB	34.7289	11.6706	0.87544	13.2380	1895.2101
HPSOB	34.7321	11.6063	0.87551	8.85148	1905.7083
SAPSOB	34.7321	11.5864	0.87552	6.3747	1908.9888
MADEPSOB	34.7325	11.6323	0.87551	8.2816	1901.4507
MDPSOB	34.7310	11.5890	0.87548	7.6127	1908.567

Table 4.8: Parameter Comparison For Test Sequence: Husky.qcif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	33.2354	0.90594	5.3921	177.7272	1025.4996
LS	33.2250	0.90574	5.3279	14.5947	1037.8485
TSS	33.2120	0.90538	5.3160	17.9789	1040.1723
NTSS	33.1858	0.90462	5.17937	16.5961	1067.6187
FSS	33.2320	0.90600	5.2933	12.9848	1044.6312
DS	33.2390	0.90585	5.3886	13.0624	1026.1569
HS	33.2466	0.90602	5.3485	9.3317	1033.8411
CDS	33.2946	0.90724	5.3505	10.1860	1033.455
KCDS	33.2557	0.90624	5.3929	8.9386	1025.3358
PSOB	33.2412	0.90606	5.3903	13.0992	1025.8341
HPSOB	33.2935	0.90693	5.3774	9.2108	1028.2986
SAPSOB	33.2805	0.90670	5.38005	5.1061	1027.7958
MADEPSOB	33.2605	0.90642	5.3816	8.2639	1027.503
MDPSOB	33.2600	0.90638	5.3824	7.2407	1027.3341

Table 4.9: Parameter Comparison For Test Sequence: Football.qcif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	35.59417	0.890876	10.54152	16.51509	572.2413
LS	35.58698	0.890481	10.61411	16.66648	568.3275
TSS	35.50216	0.887525	10.5162	18.18056	573.6189
NTSS	35.50099	0.88742	10.39848	22.47556	580.113
4SS	35.69475	0.894503	10.77881	179.1111	559.6437
DS	35.58605	0.890452	10.54836	18.60454	571.8702
HS	35.59488	0.890429	10.49248	17.69722	574.9158
CDS	35.6326	0.892028	10.53373	14.6388	572.6643
KCDS	35.58217	0.890302	10.43957	12.89991	577.8294
PSOB	35.58032	0.889875	10.18493	11.17972	592.2759
HPSOB	35.687	0.893807	10.4198	8.223519	578.9259
SAPSOB	35.6565	0.892961	10.73625	13.26269	561.8622
MADEPSOB	35.68685	0.894289	10.78577	10.56241	559.2825
MDPSOB	35.67365	0.893919	10.86549	13.80713	555.1788

Table 4.10: Parameter Comparison For Test Sequence: Soccer.qcif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	36.2919	0.86251	13.3129	177.7272	415.3539
LS	36.1679	0.85857	13.07104	18.3617	423.042
TSS	36.0617	0.85512	12.8437	18.2745	430.5288
NTSS	36.0448	0.85424	12.5729	20.5461	439.8024
FSS	36.1667	0.85853	12.8683	18.7536	429.7068
DS	36.1687	0.85849	13.0524	22.8550	423.6462
HS	36.1234	0.85656	12.69015	14.6238	435.7395
CDS	36.1731	0.85848	12.8728	24.6831	429.5547
KCDS	36.2449	0.86046	13.0242	20.2209	424.5621
HPSOB	36.2620	0.86165	13.6017	13.1963	406.5345
PSOB	36.2946	0.86233	13.2856	19.5650	416.2074
SAPSOB	36.2829	0.86294	14.0625	12.4890	393.2133
MADEPSOB	36.2483	0.8624	13.9769	14.3636	395.6214
MDPSOB	36.2575	0.86281	14.1545	15.3284	390.6576

Table 4.11: Parameter Comparison For Test Sequence: Bowing.qcif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	37.7358	0.91806	25.5572	177.7272	216.3612
LS	37.7005	0.91705	25.2416	15.3802	219.0663
TSS	37.6901	0.91672	25.2866	17.9022	218.6769
NTSS	37.6573	0.91571	24.2947	20.4367	227.6043
FSS	37.6856	0.91659	24.5798	14.2788	224.9649
DS	37.7225	0.91766	25.5875	14.7488	216.105
HS	37.6992	0.91703	25.08717	10.5071	220.4154
CDS	37.7243	0.91772	25.5303	13.9013	216.5892
KCDS	37.7324	0.91787	25.6065	10.6968	215.9451
PSOB	37.7264	0.91776	25.6501	10.3708	215.5776
HPSOB	37.7194	0.91747	25.5869	6.2346	216.1104
SAPSOB	37.7257	0.91757	25.7594	4.7995	214.6632
MADEPSOB	37.7127	0.91727	25.7719	5.7769	214.5591
MDPSOB	37.7157	0.91728	25.7018	4.5565	215.1438

Table 4.12: Parameter Comparison For Test Sequence: Tennis.sif

Test Sequences	Parameters				
	PSNR (dB)	SSIM	Compression Ratio	Search Point	Bitstream(kbps)
FS	34.5697	0.82119	11.2700	194.7090	1635.4887
LS	34.5114	0.81898	10.5313	18.0582	1750.1964
TSS	34.4754	0.81796	10.50394	19.1885	1754.769
NTSS	34.44798	0.81652	10.0339	19.8763	1836.9639
FSS	34.4972	0.81833	10.2604	17.4365	1796.4192
DS	34.5352	0.8199	11.1108	17.5717	1658.9151
HS	34.5103	0.81901	10.82635	12.45124	1702.5114
CDS	34.5427	0.81966	10.7965	16.6657	1707.2109
KCDS	34.5478	0.82027	11.0278	14.4108	1671.4041
PSOB	34.54865	0.820451	11.09175	14.13985	1661.776
HPSOB	34.53235	0.819435	10.77195	6.773242	1711.112
SAHPSOB	34.54607	0.820329	11.13795	8.904576	1654.883
MADEPSOB	34.54874	0.820431	11.1333	10.64194	1655.574
MDPSOB	34.54754	0.820337	11.18357	10.11385	1648.133

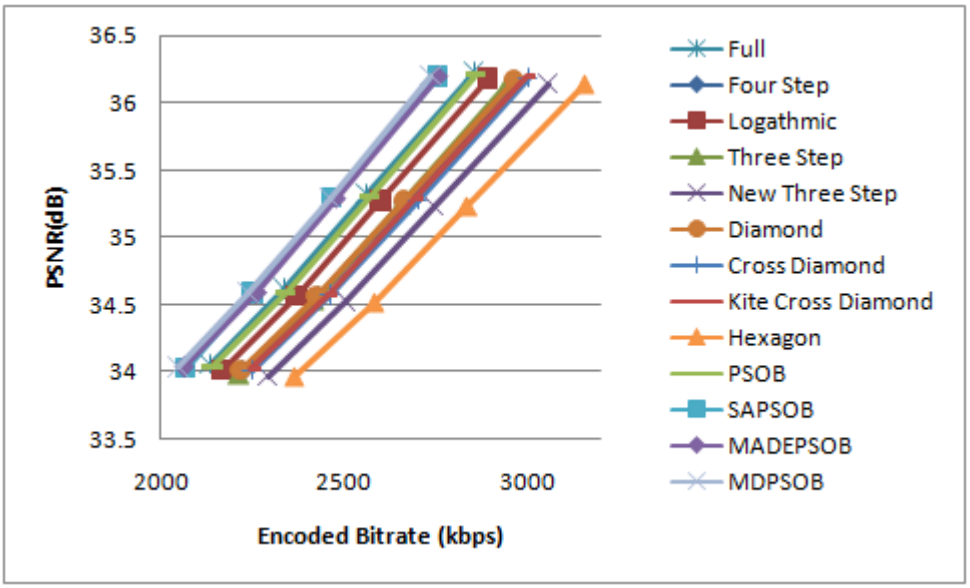


Figure 4.8: Rate Distortion Curve for Test Sequence : Stefan.cif

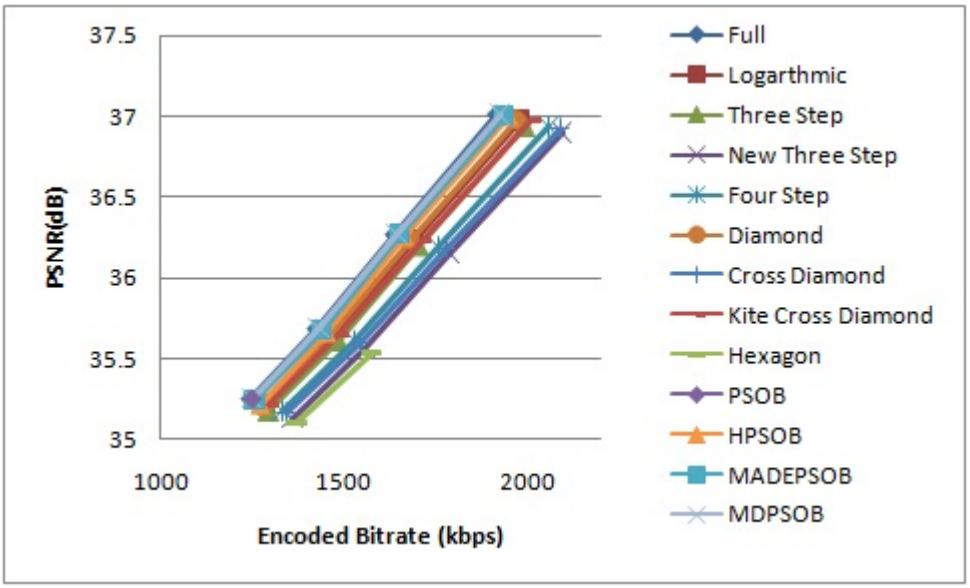


Figure 4.9: Rate Distortion Curve for Test Sequence : Soccer.cif

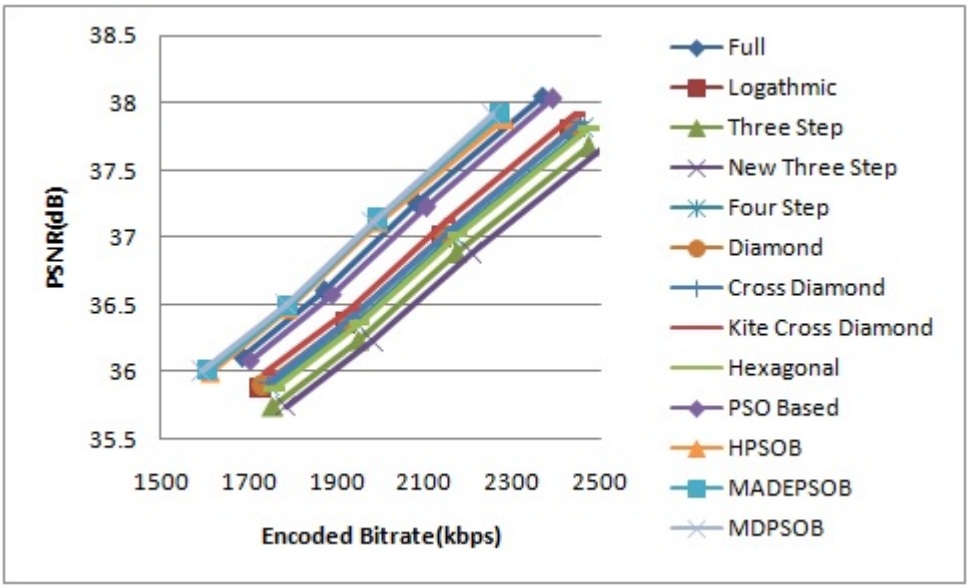


Figure 4.10: Rate Distortion Curve for Test Sequence : Football.cif

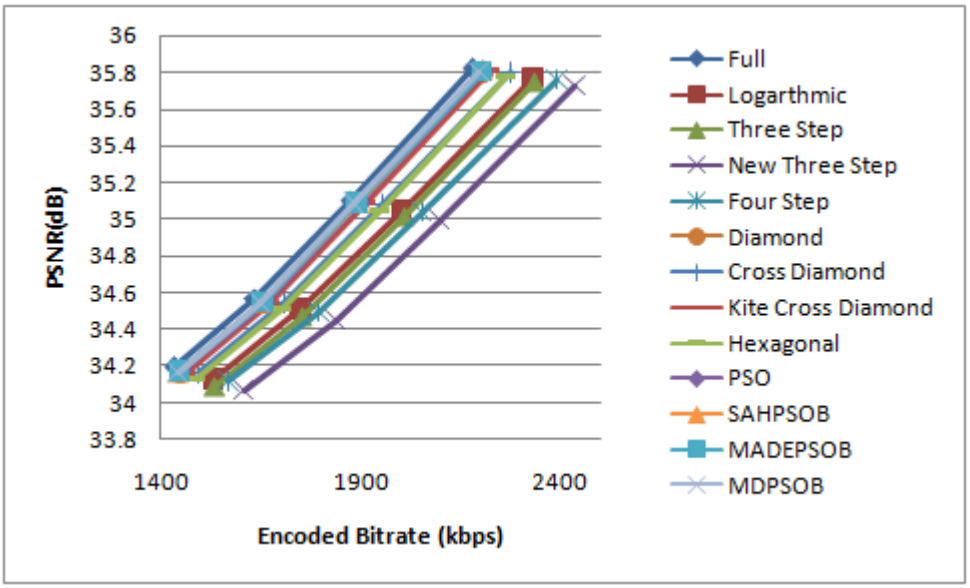


Figure 4.11: Rate Distortion Curve for Test Sequence : Tennis.cif

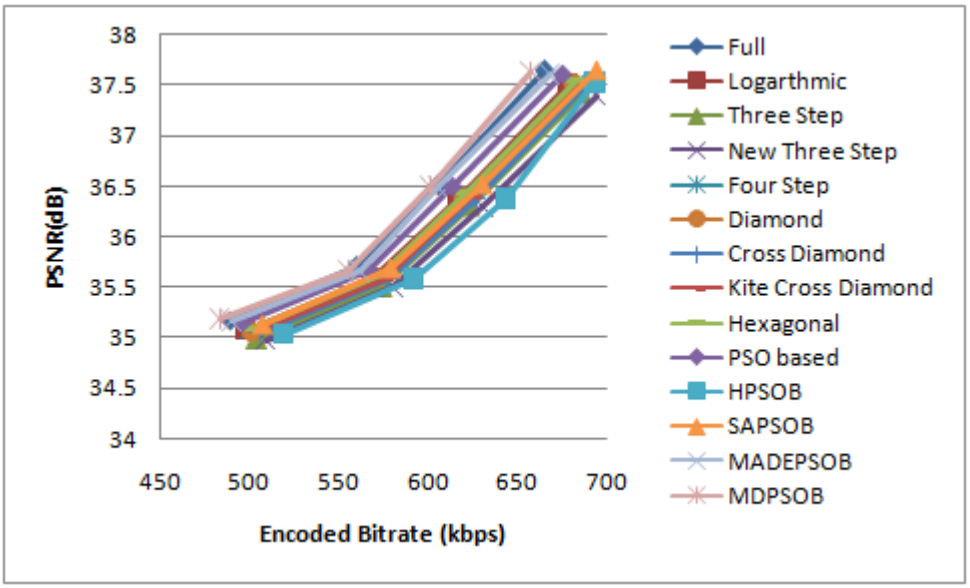


Figure 4.12: Rate Distortion Curve for Test Sequence : Football.qcif

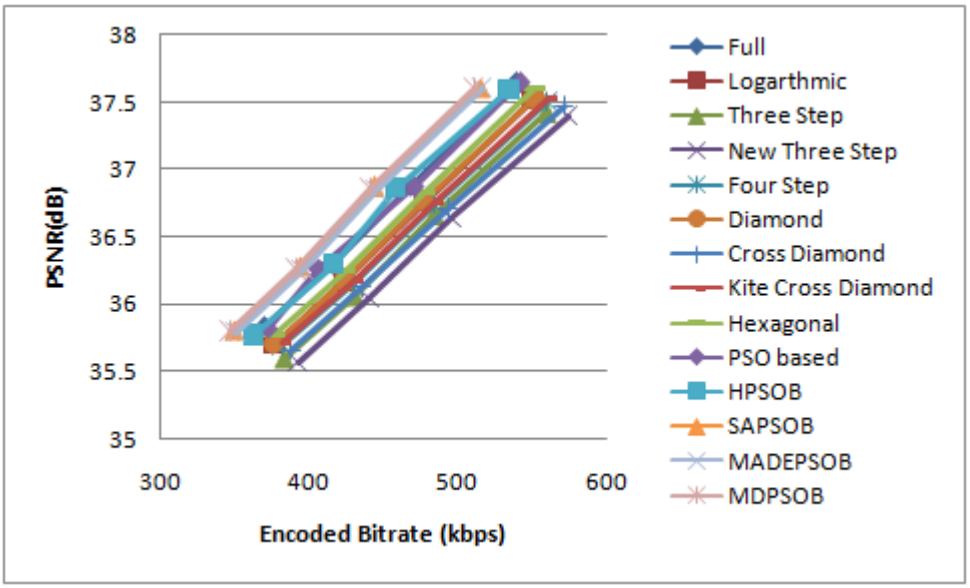


Figure 4.13: Rate Distortion Curve for Test Sequence : Soccer.qcif

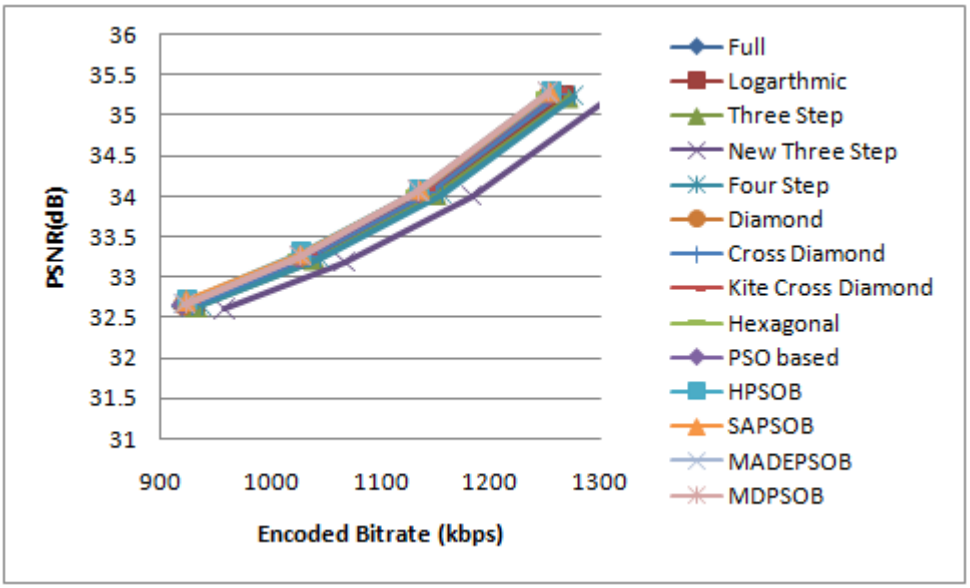


Figure 4.14: Rate Distortion Curve for Test Sequence : Husky.qcif

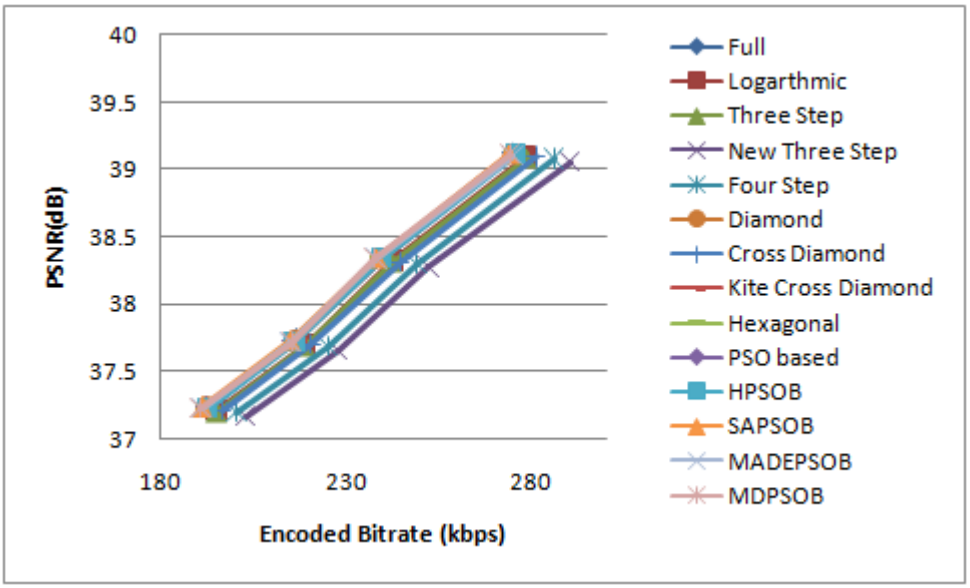


Figure 4.15: Rate Distortion Curve for Test Sequence : Bowing.qcif

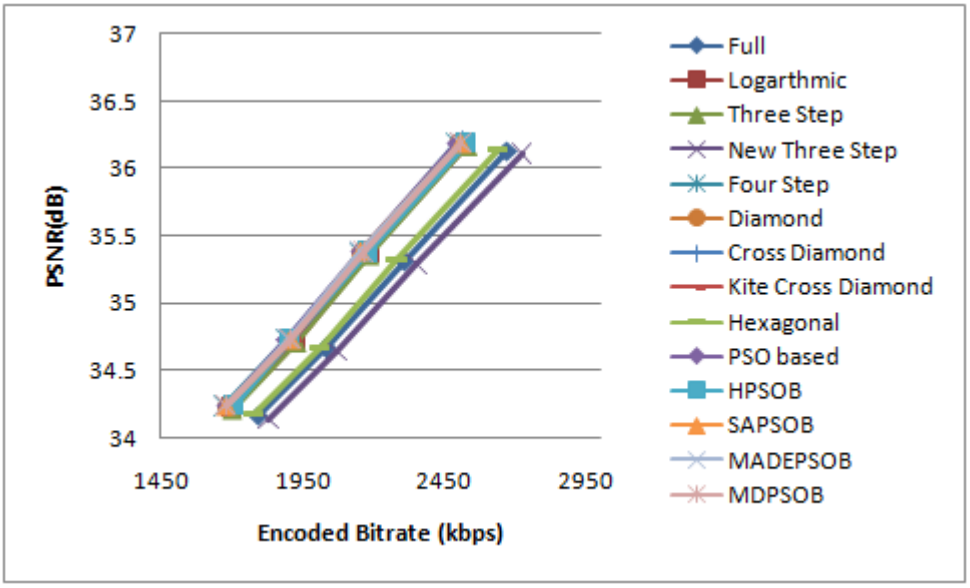


Figure 4.16: Rate Distortion Curve for Test Sequence : Coastguard.cif

4.6 Discussion

PSNR and SSIM comparison of all the test cases presented in the table shows that,they very near to Full search result or sometimes better.In almost all cases, it gives the best compression ratio after and very near to Full search. Figure 4.3, 4.6 shows the same and thus it requires less bit streams. Figure 4.4,4.5 shows it can give more compression than FS and thus require even lesser bit stream than FS. The results presented in the above section clearly shows that for each cases the proposed algorithms has the lesser search points.The computational complexity is lesser in terms of search points keeping the quality almost intact. All the Rate Distortion(RD) curve shows it mains the optimum level while changing the quantization parameter. **SAPSOB** outperforms the results with a very less number of search points.Among four proposed algorithms, **MADPSOB** and **MDPSOB** is giving best results for all the parameters sometimes with an increase of search points with respect to **SAPSOB**. So, depending on the situation the any three of this proposed algorithm can be used but overall **DAPSOB** is the best proposed algorithm which gives optimum result of PSNR, SSIM, Search Point and Compression Ratio.

Chapter 5

Proposed Up-Sampling Method

5.1 Introduction

Frame up-sampling is an important technique to produce high resolution frame from a received low resolution frame from the transmitter side at the receiver end. Generally, at the transmitting end, a video intra-frame or image is down-sampled to lessen the bandwidth required for transmission and to avoid channel congestion. At the receiver, low resolution of the down-sampled intra-frame or image is up-sampled to its original resolution by different interpolation techniques. These interpolation techniques are useful for displaying high definition standard display. In addition, interpolation plays a significant role in applications like medical diagnosis, satellite image monitoring, video surveillance and many more. In such applications, it is very often required to improve the native resolution of the original image for proper inspection and recognition.

5.2 Interpolation Techniques

5.2.1 Bilinear

There are many interpolation techniques used to up-sample the low resolution frame or image. The simple interpolation techniques among them are bilinear interpolation where the output pixel value is a weighted average of pixels in the nearest 2-by-2 neighborhood.

Though simple, Bilinear interpolation has undesirable blurring artifacts.

5.2.2 Bicubic

In Bicubic,the output pixel value is a weighted average of pixels in the nearest 4-by-4 neighborhood.

Bicubic interpolation techniques have a less blurring in compare to linear interpolation.

5.2.3 Lanczos

Lanczos is another spatial domain interpolation technique which is implemented by multiplying a sinc function with a sinc window which is scaled to be wider and truncated to zero outside of a range.

Even if Lanczos interpolation gives good results, it is slower than other approaches and provides a blurring effect in the reconstructed image.

5.2.4 DCT based

Up-sampling in DCT domain is implemented by padding zero coefficients to the high frequency side. Image resizing in DCT domain shows very good result in terms of scalability and image quality. However, this technique suffers through undesirable blurring and ringing artefacts. So there are some proposed algorithm which used DCT for Up-sampling along with some preprocessing technique.

Region Adaptive Un-Sharp masking

The region adaptive unsharp masking operation is a preprocessing step that sharpens the sub-sampled video intra frame to a certain degree depending on it's statistical local variance. The local regions with high local variance are proportionately sharpened more than the regions with less local variance by the proposed adaptive algorithm. In this operation, an unsharp mask obtained by subtracting the unsharp or smooth version of the video frame from the original. The smooth version of the video frame is obtained by

an adaptive low pass filtering operation using a region adaptive Gaussian mask whose center pixel weight is made adaptive as per the statistical local variance of a neighborhood. Furthermore, the unsharp mask is enhanced by a global scaling factor which is obtained by adding one to the global variance of the intra frame for better objective and subjective video quality. The unsharp mask thus obtained is added to the original video frame so as to obtain the sharpened video frame. The degree of sharpening obtained using the aforesaid operation compensates the extent of blurring caused by the subsequent Lanczos-3 interpolation technique and hence enhances the objective and subjective quality. The proposed method basically consists of two steps. They are namely region adaptive unsharp masking and Lanczos-3 interpolation [16]

Fuzzy Weighted Adaptive Unsharp mask based

No reference, region adaptive unsharp masking based interpolation techniques though provide good results, they lack in several aspects. Since these techniques are developed using a crisp rule, they are unable to adapt with varying constraints and thereby unable to provide better performance for different types of images and video sequences. In case of video sequences subjected to zoom in or zoom out conditions, these techniques fail drastically. Their performance also deteriorates if subjected to variation in compression ratio and video characteristics. It is because, there are only few output values for a large variation in the input values so the problems lies in the in-proper mapping between input and output values using crisp rule base[17].The problem mentioned in the previous algorithm is alleviated by using the fuzzy based mapping technique in which there will be as a de-fuzzified crisp output value corresponding to each crisp input value which may vary over a large range. Thus there is a precise and accurate mapping between input and output by using fuzzy inference system. Furthermore, this consequently improves the adaptability of the proposed fuzzy based technique with the varying conditions. Therefore, the proposed technique aims to produce very less degree of blurring and at the same time flexible enough to provide considerable performance under varying constraints such as compression ratio, video characteristics and zooming conditions. But this technique still does not reach the optimal quality both subjective and objective.

5.3 Proposed Algorithm

5.3.1 Down-Sampling in DCT Domain

To implement down-sampling in DCT domain, we take DCT of 2N × 2N image. Then we take IDCT of upper left N × N DCT coefficients to make it N × N image or intra frame.

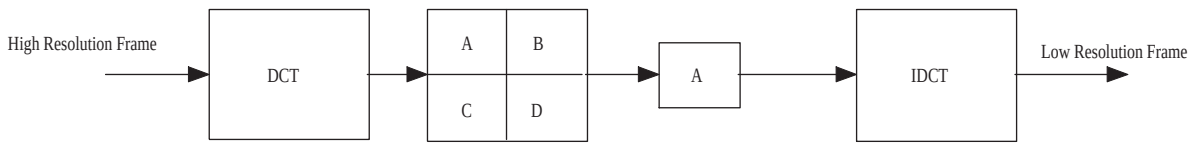


Figure 5.1: Down-sampling at transmitter

5.3.2 Wiener Filter Based Processing

Here, in the receiver spatial domain statistical approach is taken by passing the input to Wiener filter. It uses its ×x3 neighborhood to update the pixel value [11].It estimates the local mean and variance around each pixel as μ and σ^2 respectively. Wiener filter then creates a pixel wise filter using following estimates,

$$b(n1, n2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2}(a(n1, n2) - \mu) \tag{5.1}$$

where, v^2 is the average of all the local estimated variance.

Now from the equation it is clear if there is no distortion then, $v^2 = 0$ then,

$$b(n1, n2) = a(n1, n2) \tag{5.2}$$

which implies it the same as in original.

5.3.3 Up-Sampling in DCT and lanczos 3 Domain

To implement up-sampling in DCT domain, we need to add N zeros in the high frequency regions, where N is the signal length. After that, type-II IDCT of the extended 2N

samples is performed to obtain the two fold up-sampled data [7]. In case of video frames or images, the up-sampling in a matrix form is given by

$$b^{U}_{2N\times 2N} = W^{T}_{2N\times 2N} \times \begin{pmatrix} 2W_{N\times N} b_{N\times N} W^{T}_{N\times N} & 0 \\ 0 & 0 \end{pmatrix} \times W_{2N\times 2N} \tag{5.3}$$

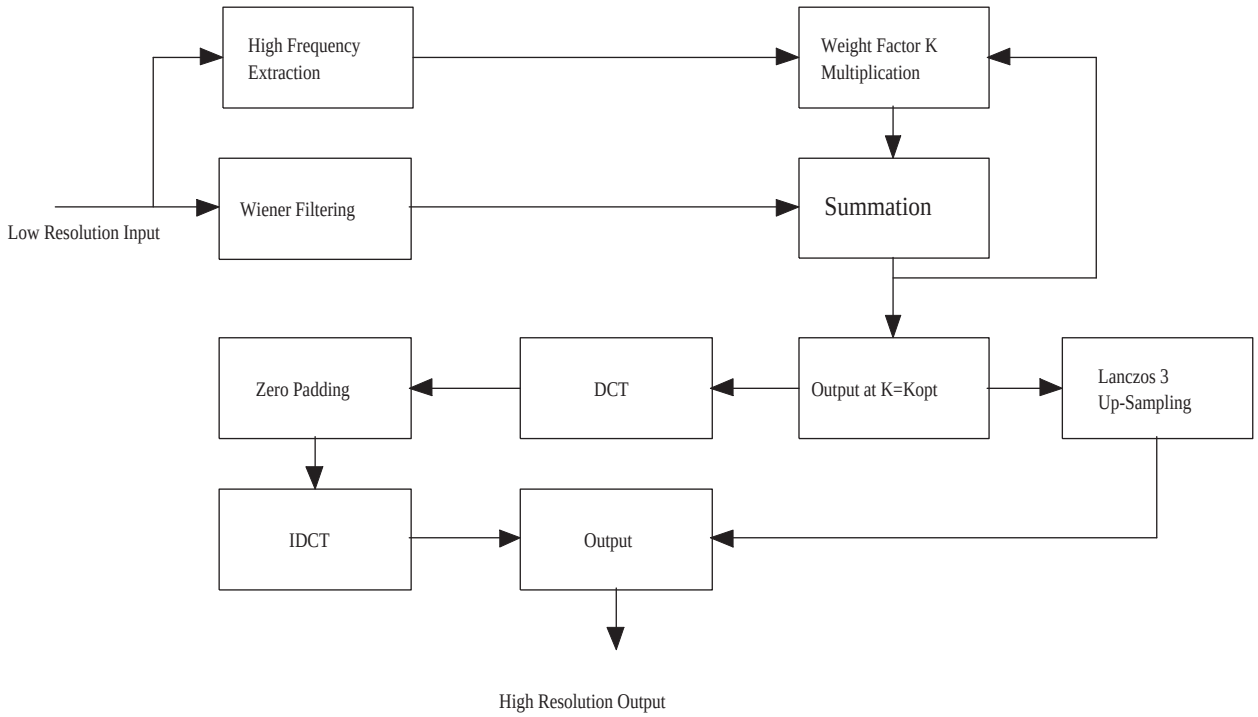


Figure 5.2: Up-Sampling at Receiver

where W denotes the 1-D type-II DCT kernel. $b_{N\times N}$ and $b^{U}_{2N\times 2N}$ are the down-sized and the up-sampled frame block. 0 is N× N zero matrix [8]. Lanczos is a spatial domain interpolation technique which is implemented by multiplying a sinc function with a sinc window which is scaled to be wider and truncated to zero outside of the main lobe. In case of Lanczos-3 interpolation, the main lobe of the sinc function along with the two subsequent side lobes on either side is used as a sinc window [1, 7]. The pixel on the interpolated values is defined by the filter's Lanczos kernel L(x). The Lanczos window is the normalized sinc function sinc(x), multiplied restricted to the main period$-a \leq x \leq a$ to form a convolution kernel for re-sampling the input field [7].

Now in our case, we have proposed output I_{out} corresponding to distorted up-sampled version I_o by the following equation (5.8)

$$I_{out} = blur + k \times (high\_frequncy) \tag{5.4}$$

which is same as

$$I_{out} = blur + k \times (I_o - blur) \tag{5.5}$$

where blur is the blur version of I_o

Now here depending on the K, there are three possibilities

Case 1 K=0: This results equation 5.8 to following

$$I_{out} = blur \tag{5.6}$$

That means the system demands to have wiener based low pass filtering. This is the case where system is noisy and for denoising is required.

Case 2: $K = 1$

$$I_{out} = blur \tag{5.7}$$

That means the system demands no per-processing rather the up-sampled version.This is the scenario when up-sampled output has better quality and no more high frequency components need to be added.

Case 3: $K < 1$

This is same as the equation 5.8 and the system demand an K times high frequency component addition for the pre processing to have the better quality output .

Case 4 : $K > 1$

This gives a very interesting result.

Let K=1+K' then from equation 5.8 we can write

$$I_{out} = blur + k' \times (I_o - blur) + (I_o - blur) \tag{5.8}$$

which is same as

$$I_{out} = blur + k' \times (I_o - blur) \tag{5.9}$$

The equation 5.9 is nothing but standard un sharpening methods. This also gives two possibilities

- **K'** > 1 High boost filtering
- **K'** < 1 Un Sharpening

5.4 Result

To compare the performance of the proposed post-processing scheme, we have taken some 11 test videos sequence and 5 images for input test signals. The input are down-sampled in the spatial domain by resizing 4:1 compression ratio and another case sampled by DCT method mentioned above. We up-sampled the frames back to their original resolution to compare with the original video frame or images. Table I and Table II illustrate the

Test Sequences	Algorithms				
	Lanczos 3	DCT	Fuzzy-Unsharp [11],[12]	Proposed I	Proposed II
Akiyo	33.4501	33.6473	33.8500	33.9526	34.7824
Bus	24.4694	24.5392	24.7582	24.8548	25.6156
City	27.8791	27.8520	27.9551	28.1085	28.6690
Coastguard	26.9391	27.0806	27.2918	27.3208	28.2468
Container	26.0086	26.2563	26.4553	26.4066	27.432
Flower	22.0333	22.0957	22.1774	22.2725	22.7767
Football	29.3665	29.687	30.0840	30.1332	31.2577
Foreman	30.9453	31.2917	31.3927	31.3762	32.3415
Hall monitor	26.8520	27.2858	27.4766	27.1846	28.6440
Ice	33.1415	33.2192	33.0547	33.5799	34.2834
Mobile	21.5967	21.7580	21.9851	22.0403	22.7815

Table 5.1: PSNR comparison of Test Video Sequence

average PSNR and SSIM comparison of DCT, Lanczos-3, Fuzzy Unsharp [12] and the proposed I Proposed II techniques for test images. Table III and Table IV illustrate the average PSNR and SSIM comparison of DCT, Lanczos-3, Fuzzy Unsharp [12] and the Proposed I Proposed II techniques for test video sequences. Experimental results reveal that for both images and video sequences Proposed II is giving best result among the four techniques with respect to PSNR and SSIM. Proposed I method is also giving better result for the most of the cases with respect to the other three methods

We have shown here PSNR comparison for two video sequences Akiyo and Foreman

Test Sequences	Algorithms				
	Lanczos 3	DCT	Fuzzy-Unsharp [11],[12]	Proposed I	Proposed II
Akiyo	0.99531	0.99553	0.99575	0.99584	0.996576
Bus	0.96059	0.96188	0.96495	0.96475	0.970512
City	0.93927	0.93871	0.94169	0.94342	0.950972
Coastguard	0.97453	0.97541	0.97694	0.97689	0.981835
Container	0.97139	0.97318	0.97489	0.97441	0.980046
Flower	0.94339	0.94439	0.94709	0.94754	0.953626
Football	0.98022	0.98182	0.98387	0.98351	0.987752
Foreman	0.99314	0.99365	0.99385	0.99379	0.9950
Hall monitor	0.97738	0.97979	0.98097	0.97944	0.98559
Ice	0.99416	0.99432	0.99412	0.99474	0.99563
Mobile	0.95657	0.95849	0.96123	0.96156	0.96826

Table 5.2: SSIM comparison of Test Video Sequence

in the graph for some defined number of frames in figure 3. It clearly shows our proposed algorithm is giving the best result.

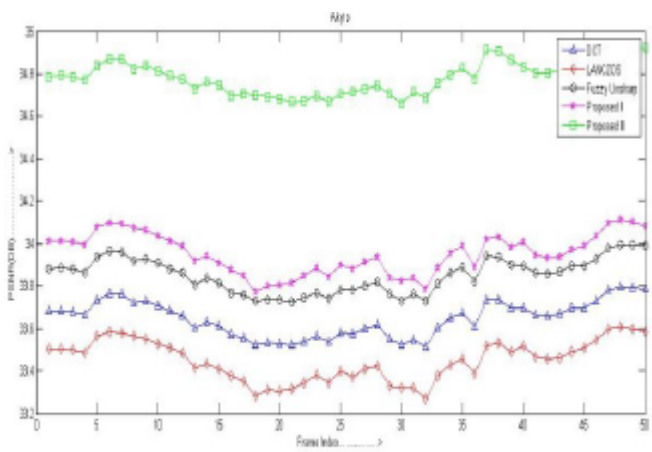


Figure 5.3: PSNR Comparison for Test Sequence : Akyio.cif

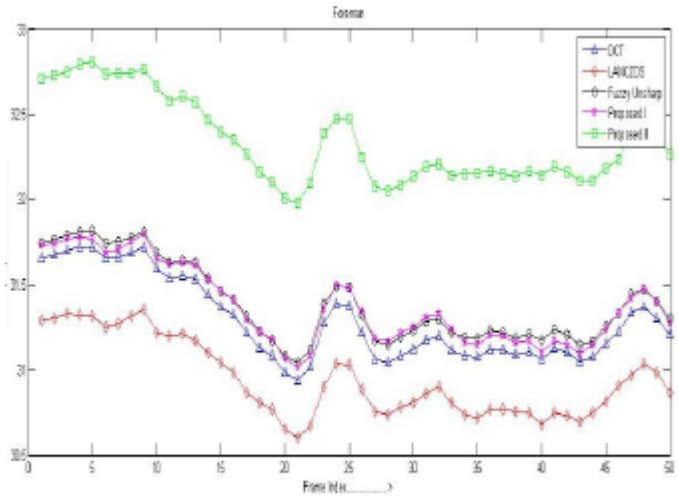


Figure 5.4: PSNR Comparison for Test Sequence : Foreman.cif

5.5 Discussion

The proposed schemes are an approach to restore the lost information during the down sampling and up-sampling operation. It restores the fine details and edge information of video and image. In addition, this overcomes the problem of blurring artifacts caused by such operation. The improvement is gained by using the local statistics based Wiener filtering with statistical local variance of a neighbourhood on direct mapping basis. In addition, the proposed scheme performs quite adaptively under various constraints such as change in compression ratio (i.e. 16:1). This method adaptively removes noise content added to the image or video frame. Thus this algorithm is giving good qualitative and subjective quality output.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The proposed algorithms are based on PSO and DE with the knowledge of spatial and temporal motion vector. We compared our proposed method with conventional motion estimation algorithms for many other test video sequences . The experimental result shows that our proposed algorithms are efficient to maintain the accuracy. It gives the best optimized result PSNR, SSIM, Compression ratio, Encoded Bitrate. Rate distortion curve shows that it matches the optimum level without degrading the quality in terms of PSNR. Apart from all these there is huge improvement in computational complexity in terms of search points. Moreover, the good subjective quality can be visualized in terms of sharpened edge; less degree of blurring and fine details preservation. Thus this algorithm is giving good quality, high accuracy, high speed and less computational complexity output.As discussed already **MDPSOB** is the best proposed algorithm which gives an optimized results of the all requirements. The proposed schemes in up-sampling are an approach to restore the lost information during the down sampling and up-sampling operation. It restores the fine details and edge information of video and image. In addition, this overcomes the problem of blurring artifacts caused by such operation. The improvement is gained by using the local statistics based Wiener filtering with statistical local variance of a neighborhood. In addition, the proposed scheme performs quite adaptively under various constraints such as change in compression ratio (i.e. 16:1). This method

adaptively removes noise content added to the image or video frame. Thus this algorithm is giving good qualitative and subjective quality output.

6.2 Future Work

Motion estimation algorithm proposed in this thesis is based on two evolutionary model PSO and DE.The tempral and spatial correlation information used for faster optimization. The proposed algorithm is adaptive to the type and direction of motion.But still there needs some improvements. So, future work can be a model of motion based on the dynamics of the motion body or object moving.For up-sampling, research still demands and thus future work is designing of a new interpolation technique with better degree of blurring and ringing artifacts . It should be faster and have less burden for optimization of parameter.

Bibliography

[1] R. C. Gonzalez, *Digital image processing.* Pearson Education India, 2009.

[2] I. E. Richardson, *Video codec design: developing image and video compression systems.* John Wiley & Sons, 2002.

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

[4] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," *Communications, IEEE Transactions on*, vol. 29, no. 12, pp. 1799–1808, 1981.

[5] T. Koga, "Motion-compensated interframe coding for video conferencing," in *Proc. NTC 81*, pp. C9–6, 1981.

[6] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 4, no. 4, pp. 438–442, 1994.

[7] 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.

[8] 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.

[9] 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.

[10] 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, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, vol. 3, pp. III–729, IEEE, 2004.

[11] C. Zhu, X. Lin, and L.-P. Chau, "Hexagon-based search pattern for fast block motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 5, pp. 349–355, 2002.

[12] 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.

[13] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1, pp. 39–43, New York, NY, 1995.

[14] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of Machine Learning*, pp. 760–766, Springer, 2010.

[15] J. Cai and W. D. Pan, "On fast and accurate block-based motion estimation algorithms using particle swarm optimization," *Information Sciences*, vol. 197, pp. 53–64, 2012.

[16] A. Acharya and S. Meher, "Region adaptive unsharp masking based lanczos-3 interpolation for video intra frame up-sampling," in *Sensing Technology (ICST), 2012 Sixth International Conference on*, pp. 57–62, IEEE, 2012.

[17] A. Acharya and S. Meher, "No reference, fuzzy weighted unsharp masking based dct interpolation for better 2-d up-sampling," in *Fuzzy Systems (FUZZ), 2013 IEEE International Conference on*, pp. 1–8, IEEE, 2013.

[18] A. C. Bovik, *Handbook of image and video processing.* Academic press, 2010.

[19] K. Bakwad, S. Pattnaik, B. Sohi, S. Devi, S. Gollapudi, C. V. Sagar, and P. Patra, "Small population based modified parallel particle swarm optimization for motion estimation," in *Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on*, pp. 367–373, IEEE, 2008.

[20] X. Yuan and X. Shen, "Block matching algorithm based on particle swarm optimization for motion estimation," in *Embedded Software and Systems, 2008. ICESS'08. International Conference on*, pp. 191–195, IEEE, 2008.

[21] G.-Y. Du, T.-S. Huang, L.-X. Song, and B.-J. Zhao, "A novel fast motion estimation method based on particle swarm optimization," in *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*, vol. 8, pp. 5038–5042, IEEE, 2005.

[22] R. Ren, Y. Shi, B. Zheng, *et al.*, "A fast block matching algorithm for video motion estimation based on particle swarm optimization and motion prejudgment," *arXiv preprint cs/0609131*, 2006.

[23] *International Telecommunication Union ITU-T Recommendation H.262.* 1996.

[24] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Evolutionary Computation, 2004. CEC2004. Congress on*, vol. 2, pp. 1980–1987, IEEE, 2004.

[25] S.-Z. Zhao, P. N. Suganthan, and S. Das, "Self-adaptive differential evolution with multi-trajectory search for large-scale optimization," *Soft Computing*, vol. 15, no. 11, pp. 2175–2185, 2011.

[26] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: A comparative study on numerical benchmark problems," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 6, pp. 646–657, 2006.

[27] S. Das and P. N. Suganthan, "Differential evolution: a survey of the state-of-the-art," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 4–31, 2011.

[28] E. Cuevas, D. Zaldivar, M. Pérez-Cisneros, and D. Oliva, "Block-matching algorithm based on differential evolution for motion estimation," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 1, pp. 488–498, 2013.

[29] R. Keys, "Cubic convolution interpolation for digital image processing," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 6, pp. 1153–1160, 1981.

[30] S. E. Reichenbach and F. Geng, "Two-dimensional cubic convolution," *Image Processing, IEEE Transactions on*, vol. 12, no. 8, pp. 857–865, 2003.

[31] W. Ye and A. Entezari, "A geometric construction of multivariate sinc functions," *Image Processing, IEEE Transactions on*, vol. 21, no. 6, pp. 2969–2979, 2012.

[32] R. Dugad and N. Ahuja, "A fast scheme for image size change in the compressed domain," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 4, pp. 461–474, 2001.

[33] Z. Wu, H. Yu, and C. W. Chen, "A new hybrid dct-wiener-based interpolation scheme for video intra frame up-sampling," *Signal Processing Letters, IEEE*, vol. 17, no. 10, pp. 827–830, 2010.

[34] 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*, pp. 47–73, Springer, 2014.

List of Publications

1. Dhara, S.K, and Meher, S "A Novel Hybrid Technique in Spatio-Frequency Domain for Better Quality Image and Video Up-Sampling" *India Conference (INDICON),2014 Annual IEEE*. IEEE 2014

2. Dhara, S.K, Meher, S, Singh D "A Novel Block Matching Based Motion Compensation Using Hybrid Particle Swarm Optimization Technique for Efficient Video Compression'*In Signal Processing and Information Technology (ISSPIT). 2014 IEEE International Symposium on.* IEEE, 2014