

MOTION ESTIMATION FOR SCALABLE VIDEO CODING

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree*

of

MASTER OF TECHNOLOGY

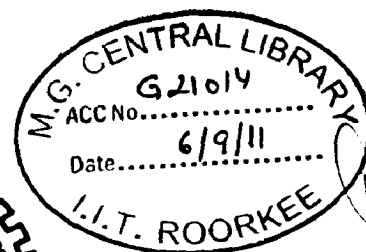
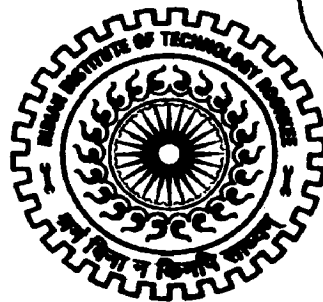
in

ELECTRONICS AND COMMUNICATION ENGINEERING

(With Specialization in Wireless Communication)

By

YOGESHWAR SINGH



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
JUNE, 2011**

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report entitled, "MOTION ESTIMATION FOR SCALABLE VIDEO CODING" towards the partial fulfilment of the requirements for the award of the degree of **Master of Technology** with specialization in **Wireless Communications**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology, Roorkee (INDIA) is an authentic record of my own work carried out during the period from July 2010 to June 2011, under the guidance of **Dr. DEBASHIS GHOSH, Associate Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee.**

I have not submitted the matter embodied in this dissertation for the award of any other Degree or Diploma.

Date: 03-06-2011

Place: Roorkee



YOGESHWAR SINGH

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: 03-06-2011

Place: Roorkee

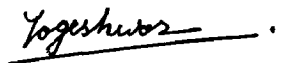

Dr. Debashis Ghosh,
Associate Professor,
E & C Department,
IIT Roorkee,
Roorkee-247667(INDIA)

ACKNOWLEDGEMENTS

First of all, I would like to express my deep sense of respect and gratitude towards my guide **Prof. Debashis Ghosh**, who has been the guiding force behind this work. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I would like to thank all faculty members and staff of the Department of Electronics and Computer Engineering, IIT Roorkee for their generous help in various ways for the completion of this thesis.

I am greatly indebted to all my batch mates, who have helped me with ample moral support and valuable suggestions. Most of all I would like to thank my family. Finally, I would like to extend my gratitude to all those persons who directly or indirectly contributed towards this work.



YOGESHWAR SINGH

LIST OF ABBREVIATIONS

ABME	Activity Based Motion Estimation
BDM	Block Distortion Measure
BL	Base Layer
BR	Boundary Region
CIF	Common Intermediate Format, a colour image format
CODEC	<i>CO</i> der / <i>DE</i> Coder pair
EL	Enhancement Layer
GOP	Group Of Pictures, a set of coded video images
HPA	Half-Pixel Accuracy
ILAM	Inter Layer Activity Model
IPA	Integer Pixel Accuracy
ISO	International Standards Organisation, a standards body
IR	Interior Region
ITU	International Telecommunication Union, a standards body
MB	Macro Block
ME	Motion Estimation
MV	Motion Vector
MVD	Motion Vector Difference

MVP	Motion Vector Predictor
MSE	Mean of Squared Error
MPC	Matching Pel count
MPEG	Motion Picture Experts Group, a committee of ISO/IEC
PSNR	Peak Signal to Noise Ratio, an objective quality measure
QCIF	Quarter Common Intermediate Format
QPA	Quarter Pixel Accuracy
SAD	Sum of Absolute Difference
SR	Search Range
SSA	Sub-Sample Accuracy
SVC	Scalable Video Coding

ABSTRACT

Motion estimation is important component of video coding systems because it enables us to exploit the temporal redundancy in the video sequence. Generally, block based motion estimation algorithms are used which search the macro-blocks in the reference frames for the best match in the vicinity of the location of current macro-block. Estimation can further be improved by searching for matching macro-blocks at sub-pixel positions in the reference frame. Motion estimation is a computationally expensive process and the complexity increases as the sub-pixel resolution of motion vectors is increased.

For encoding spatially scalable video, motion estimation process is required to be performed for each of spatial resolution. Therefore, such techniques are required for motion estimation for scalable video coding which are less computationally expensive and also provide good PSNR performance. Activity based motion estimation technique which has been proposed recently, dynamically adapts the search range of motion estimation in enhancement layer based upon the activity of corresponding macro-block in base layers.

In this thesis we have proposed a new motion estimation scheme which decreases the overall complexity of activity based motion estimation scheme while maintaining similar PSNR performance.

Table of Contents

Chapter 1	Introduction	1
1.1	Video Coding	1
1.2	Scalable Video Coding	1
1.2.1	Types of Scalability	2
1.3	Scalability in Existing Video Coding Standards.....	3
1.4	Motivation.....	5
1.5	Thesis Organization	6
Chapter 2	Fundamental Concepts of Motion Estimation.....	7
2.1	Block Matching Algorithm	7
2.1.2	Forward Motion Estimation.....	9
2.1.3	Bi-Directional Motion Estimation	9
2.1.4	Block Matching Criteria for Block Matching Algorithm.....	11
2.1.5	Block Size.....	13
2.1.6	Search Range	14
2.1.7	Search Accuracy	14
2.2	Fast Block Matching Motion Estimation Algorithms.....	15
2.2.1	Three Step Search (TSS)	15
2.2.2	Four Step Search (4SS).....	16
2.2.3	Diamond Search (DS).....	17
2.2.4	Comparison of Fast Block Matching Motion Estimation Algorithms	18
Chapter 3	Motion Estimation with Sub-Sample Accuracy (SSA).....	20
3.1	Conventional Sub-Sample Accurate Motion Estimation (SSA ME).....	20
3.2	Interpolation Free Sub-pixel Accurate Motion Estimation.....	23
3.2.1	Model Description	23
3.2.2	Parameter estimation	24
3.2.3	Results of Interpolation-Free Sub-pixel Accurate Motion Estimation.....	25

Chapter 4	Motion Estimation for Scalable Video Coding (SVC).....	28
4.1	Introduction to Scalable Video Coding (SVC)	28
4.2	Activity-Based Motion Estimation Scheme for Scalable Video Coding.....	29
4.2.1	Spatial Scalability.....	29
4.2.2	Base Layer Motion Vector Predictor (BL MVP)	30
4.2.3	Concept of Activity	32
4.2.4	Inter Layer Activity Model.....	33
4.2.5	Procedure for simulation of Activity Based Motion Estimation (ABME) scheme	35
4.2.6	Simulation Results for Activity Based Motion Estimation scheme	37
Chapter 5	Proposed Motion Estimation Scheme for Scalable Video Coding.....	39
5.1	Proposed Motion Estimation Scheme	39
5.2	Simulation Results and Observations	40
Chapter 6	Conclusion	46
Works Cited	47

List of Figures

Figure 1-1 General principle of Scalable Video Coding having T layers	4
Figure 1-2 Spatial Scalability in a hybrid coder: Double loop, supporting switchable Motion Compensated (MC) prediction in the enhancement layer	5
Figure 2-1 General principle of Motion Estimation	7
Figure 2-2 Motion Estimation using Block Matching Algorithm.....	8
Figure 2-3: Backward Motion estimation with current frame as k and frame $(k-1)$ as the reference frame.	8
Figure 2-4 Forward Motion estimation with current frame as k and frame $(k+1)$ as the reference frame.	9
Figure 2-5 A video sequence showing the benefits of bi-directional prediction.....	10
Figure 2-6 MPEG group of pictures.	10
Figure 2-7 Three step search	16
Figure 2-8 Search patterns of the 4SS. (a) First step, (b) intermediate step, (c) alternate step, and (d) last step.....	16
Figure 2-9 Diamond Search algorithm, (a) The corner point (LDSP->LDSP) (b) The edge point (LDSP->LDSP)(c) The centre point(LDSP->SDSP)	17
Figure 2-10 Comparative PSNR performance for ICE video sequence, at QCIF resolution, 15 fps for ES, TSS, 4SS and DS	18
Figure 2-11 Comparison of Number of BDM computations per MB for ICE video sequence, at QCIF resolution, 15 fps for TSS, 4SS and DS	19
Figure 3-1: Plot of PSNR for Integer pixel Accuracy Motion estimation and Compensation Vs PSNR for Conventional Half pixel Accurate Motion estimation and compensation. Video sequence: Soccer, 15 fps, Format: QCIF, Macro-block Size: 16.	21
Figure 3-2: Position of neighbouring macro-blocks.....	23
Figure 3-3: Comparison of Interpolation-free Half -Pixel Accurate Motion Estimation (HPA ME), Conventional half-pixel Motion Estimation and Integer pixel Motion Estimation. : Soccer, Format: QCIF, Macro-block Size: 16, search parameter: 8.	26
Figure 4-1 MV^{l-1} , which denotes the final MV at the base layer (layer $l - 1$), is used to obtain MVs^l , i.e., the base-layer motion vector predictor of the corresponding MBs at the enhancement layer (layer l).	30
Figure 4-2 Length of the longer edge of the MBB, L_{MBB} where (x_s, y_s) denotes the BL MVP, and (x, y) denotes the final motion vector.	30
Figure 4-3 Cumulative distribution of L_{MBB} , the maximum length of MBB, for video sequences: Harbour, Soccer, Ice and Crew. QCIF and CIF as base layer and enhancement layer respectively.....	31
Figure 4-4 Grid on a sample object in (a) base layer and (b) enhancement layer; the rectangles with a bold line, B0 and C0-3, denote 4×4 blocks in the corresponding positions in the base layer and the enhancement layer, respectively.	32
Figure 4-5 Activity plane representing pairs of the mean of activities between two neighbouring layers [base layer (BL) = CIF, enhancement layer (EL) = 4CIF] for video sequence ICE. The dashed line denotes inter-	

layer activity model with the given slope, $\alpha (= \Delta A_l / \Delta A_l - 1)$ and the given intercept, θ where $A_l - 1$ and A_l denote the mean of activities over all MBs in a frame at the base and enhancement layer, respectively..... 34

Figure 4-6 Optional check of diamond-shaped points (OCDSP), where SR is derived from (4.2). Point "I" denotes the point with the minimum SAD cost within the given search range, d denotes the distance between "Predictor" and point I, and point "J" denotes the centre point of the diamond-shaped search pattern whose distance from "Predictor" is twice as long as d . Five gray coloured circles denote optional check points in the diamond-shaped search pattern, and point "K" denotes the point with the minimum SAD cost among six candidates, i.e., five optional check points and point I. SR^{new} denotes the required search range to cover point K (a) when point K is different from point I, and (b) when point K is identical to point I. 36

Figure 5-1 Comparison of PSNR obtained for BL by simulation of proposed ME scheme with that obtained by the simulation of ABME scheme, Video Sequence: Soccer, 15 fps. 41

Figure 5-2 Comparison of PSNR obtained for EL1 by simulation of proposed ME scheme with that obtained by the simulation of ABME scheme, Video Sequence: Soccer, 15 fps. 42

Figure 5-3 PSNR Gain obtained for EL2 by simulation of proposed ME scheme over the ABME scheme, Video Sequence: Soccer, 15 fps. 42

Figure 5-4 Comparison of number of computations of Block Distortion Measure (BDM) per MB for EL2 obtained using the proposed scheme with that of ABME scheme, Video Sequence: Soccer, 15 fps. 43

Figure 5-5 Comparison of total CPU time for Proposed scheme with that of ABME scheme, Video Sequence: Soccer, 15 fps. 44

List of Tables

Table 3-1: Comparison of Conventional Half-Pixel Accurate (HPA) motion estimation with Integer-Pixel Accurate (IPA) motion estimation based on PSNR, average increase in CPU time (ΔT) and increase in number of computations of Block Distortion Measure per Macro-Block (Δ CBDM/MB).....	22
Table 3-2: Comparison of Interpolation-free Half-Pixel Accurate (HPA) motion estimation with Integer-Pixel Accurate (IPA) motion estimation based on PSNR, average increase in CPU time (ΔT) and increase in number of computations of Block Distortion Measure per Macro-Block (Δ CBDM/MB).....	27
Table 4-1 Inter-Layer Activity Prediction Factor, α, and Inter-Layer Activity Prediction Offset, β, for the given sequence, and the root mean square error, for given α and β, measured with generic sequences (45 frames on a SVC structure comprising three layers)	35
Table 4-2 Results as obtained through simulation of Activity Based Motion Estimation (ABME) method. PSNR for BL, EL1 and EL2 along with number of Computations of Block Distortion Measure per macro-block CBDM/MB for EL2 and total CPU Time (T) consumed during simulation for evaluation of motion vectors for all three layers.....	38
Table 5-1 Comparison of Proposed ME scheme with ABME scheme based on PSNR at all layers, number of computations of BDM per MB (CBDM/MB) and CPU time (T)	45

Chapter 1 Introduction

1.1 Video Coding

Digital video coding has gradually increased in importance since the 90s when MPEG-1 first emerged. It achieves higher data compression without significant loss of subjective video quality. The volume of digital video data is very large making it impossible to transmit raw video data through communication channels of limited transmission bandwidth or to save them on storage devices. This calls for data compression that provides efficient transmission or storage. Compression addresses the problem of reducing the amount of data required to represent a digital image. Generally speaking, video compression is a technology for transforming video signals that aims to retain original quality under a number of constraints, e.g. storage constraint or computation power constraint. It takes advantage of data redundancy between successive frames to reduce the storage requirement by applying computational resources. The design of data compression systems normally involves a trade-off between quality, speed, resource utilization and power consumption.

In a video scene, data redundancy arises from spatial, temporal and statistical correlation between frames. These correlations are processed separately because of differences in their characteristics. Hybrid video coding architectures have been employed since the first generation of video coding standards, i.e. MPEG. It consists of three main parts to reduce data redundancy from the three sources described above. Motion estimation and compensation are used to reduce temporal redundancy between successive frames in the time domain. Transform coding, also commonly used in image compression, is employed to reduce spatial dependency within a frame in the spatial domain. Entropy coding is used to reduce statistical redundancy over the residue and compression data.

1.2 Scalable Video Coding

Merely a compression scheme may not be solution to some application such as image database browsing, video-on-demand and video communication over heterogeneous networks. In these situations, other properties such as scaling for a desired spatial resolution, frame rate, bit rate etc are fast becoming indispensable features for a good and comprehensive compression system. Different users may have different requirements and limitations on bit rate, display resolution, frame rate and decoding complexity which cannot be anticipated in advance during compression.

1.2.1 Types of Scalability

Digital video, as a multidimensional signal allows many possible specifications such as picture quality, picture size and picture playback rate. The ability to scale and choose different combinations of these video specifications is crucial for simultaneous distribution to separate clients. These scalable video features are denoted as *video scaling parameters*. These desirable video scaling parameters include spatial resolution, temporal resolution, SNR, bit rate and complexity.

1. **Spatial Resolution Scalability:** It refers to the flexibility to support different display resolutions by selecting subsets of a common compressed video bit stream. It allows playback of same video on various display devices with separate display resolutions.
2. **Temporal Resolution Scalability:** It enables the flexibility to choose different video frame rates for playback from a common compressed video source. A higher frame rate will allow smooth motion rendition, while a lower frame rate causes perception of jerkiness. Scaling of video playback frame rate is in fact one of the best choices for bit rate scalability while preserving average video quality level.
3. **SNR Scalability:** It enables the selection of different video qualities for common compressed video bit stream. Generally, video quality improves as more video data are used to reconstruct the video. As a result there is an intrinsic one to one relationship between video bit rate scalability and SNR scalability, if other scaling parameters remain unchanged.
4. **Bit Rate Scalability:** It allows either source or consumer to gracefully scale for wide range of different data rates from same scalable video source. Due to constraints in effective network bandwidth, it is very desirable to have constant bit rate videos. However, it usually results in intermittent fluctuation of video quality around frames with high motion content.
5. **Complexity Scalability:** It refers to trade-off between the computational complexity and rate-distortion performance of the video encoder/decoder by means of scaling for different subsets of same compressed video bit stream. Since real-time encoding and decoding is critical in many video applications, a scalable video coding algorithm should provide the flexibility to the lower end processors to scale down the video frame rate, for instance, in order to maintain real time encoding and decoding.

1.3 Scalability in Existing Video Coding Standards

Early video compression standards such as ITU-T H.261 [1] and ISO/IEC MPEG-1 [2] did not provide any scalability mechanisms. One reason for this was the dedicated design for specific applications such as conversational services or storage, which did not require scalability. In fact, scalability can nevertheless be achieved by providing different bit streams targeting at different decoded resolutions. This method is called *simulcast* which ties together two or several streams for the purpose of parallel transmission or parallel storage. ISO/IEC MPEG-2 [3], which is identical to ITU-T H.262, was the first general-purpose video compression standard which also includes a number of tools providing scalability. MPEG-2 was the first standard to include implementations of *layered coding*, where the standalone availability of enhancement information (without the base layer) is useless, because differential encoding is performed with reference to the base layer. It supports spatial, temporal and SNR scalability however, the number of scalable bit stream layers is generally restricted to a maximum of three in any of the existing MPEG-2 profiles.

The video codec of the ISO/IEC MPEG-4 standard [4] provides even more flexible scalability tools, including spatial and temporal scalability within a more generic framework, but also SNR scalability with fine granularity and scalability at the level of (eventually semantic) video objects. Advanced Video Coding, as defined as part 10 of the MPEG-4 standard [5], aka ITU-T H.264, can in principle be run in different temporal scalability modes, due to its flexibility in the definition of prediction frame references.

Nevertheless, it must be noted that any of the video coding standards existing so far restricts scalability at the bit stream level to a predefined number of *layers* which must be known at the time of encoding.

A very general principle of (layered) scalable coding and decoding is shown in Figure 1-1, where by supplementing further building blocks of the intermediate-level type (highlighted by a dotted rectangle), an arbitrary number of scalable layers can in principle be realized. The spatiotemporal signal resolution to be represented by the base layer is first generated by decimation (pre-processing). The *mid-processing* unit performs up-sampling of the next lower layer signal to the subsequent layer's resolution.

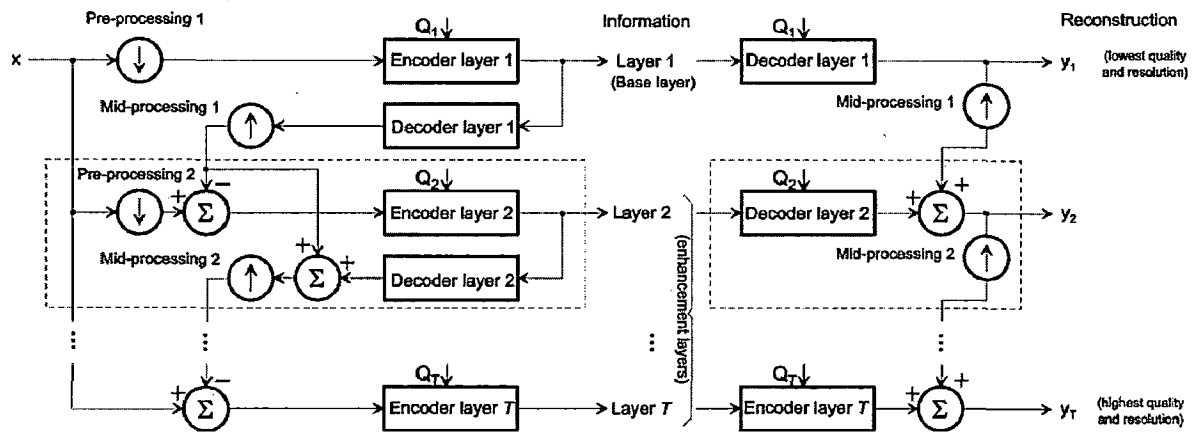


Figure 1-1 General principle of Scalable Video Coding having T layers

The information is propagated from the lower into the higher resolution layers both during encoding and decoding. The base layer and any composition from layers should in the ideal case be self-contained, which means that the prediction should not use any decoded information from higher layers. Otherwise, different estimates would be used at the encoder and decoder sides, and a *drift effect* would occur [6]. As the base-layer information is used for prediction of the enhancement layer, the rate-distortion performance toward higher rates will be worse than it could be in a single-layer coder which may dramatically affect the overall compression performance when the base-layer quality is low.

Alternatively, the full enhancement information could blindly be used for prediction; in this case, the reconstruction quality of the highest enhancement layer approaches the performance of a single-layer coder, while the reconstruction quality of the base layer and all intermediate layers would eventually suffer dramatically due to the drift. This basic dilemma to *penalize either enhancement or base-layer* performance is inherently caused by the recursive frame prediction nature of the hybrid coding concept.

To achieve higher compression performance, inter-frame prediction with a separate loop can be applied to the enhancement layer coding as shown in figure 1-2. This will nevertheless still provide a worse rate-distortion performance than a single-layer codec where the full reconstructed information can be used in a single prediction loop. However, it is possible in a similar double-loop method to track the drift within the local loop of the encoder that would occur in a decoder only receiving the base-layer information.

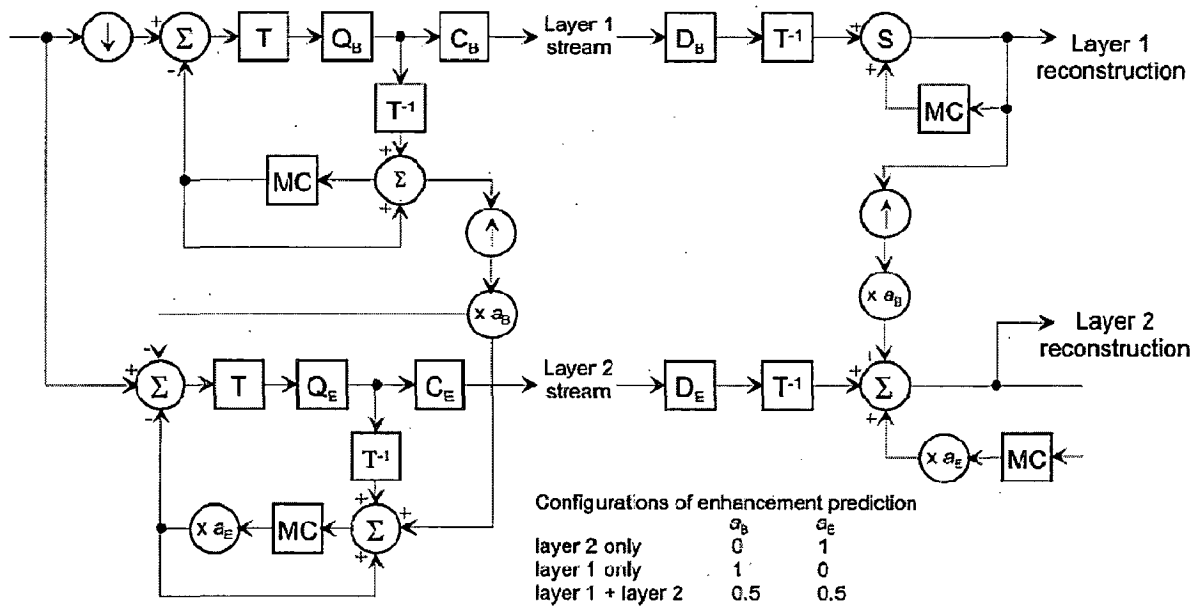


Figure 1-2 Spatial Scalability in a hybrid coder: Double loop, supporting switchable Motion Compensated (MC) prediction in the enhancement layer

For spatial scalability however, there may be cases where using only the previous frame enhancement layer reconstruction allows better prediction of the actual enhancement frame, without referencing the current base-layer frame. Such a more flexible structure is depicted in Figure 1-2. Here, the enhancement layer frame can either be predicted entirely from the up-sampled base layer, from the previous enhancement layer reconstruction, or from the mean value of both.

1.4 Motivation

In the implementations of the modern video encoders supporting spatial scalability as shown in figure 1-2 it is required to perform motion estimation separately for each of the spatial resolution. Motion estimation process for various layers becomes increasingly computationally expensive as the resolution increases. Motion vectors for these spatially separate layers are highly correlated with each other. Hence, various motion estimation schemes have been developed to take advantage of the correlation between the motion vectors of these distinct motion estimation processes. These motion estimation schemes aims to reduce the computational complexity of these motion estimation processes while maintaining the estimation quality.

In our work we have tried to decrease the computational complexity of these motion estimation processes further while maintaining almost similar estimation quality.

1.5 Thesis Organization

In Chapter 2 we present the concept of block based motion estimation. In this we include different classification of motion estimation technique depending on the choice of reference frames. Various types of Block Distortion Measures which are generally used to compute the quality of *match* are presented. Fast block matching motion estimation algorithms are presented and their performances are compared.

In Chapter 3 we discuss sub-pixel accurate motion estimation and compare the computational complexity and PSNR performance with integer pixel accurate motion estimation. A fast interpolation-free sub-pixel accurate motion estimation scheme is presented and its performance is compared with conventional sub-pixel accurate motion estimation scheme.

In Chapter 4 an Activity Based Motion Estimation Scheme for Scalable Video Coding is discussed and its performance in terms for computational complexity and PSNR performance is evaluated.

In Chapter 5 we propose a motion estimation scheme and compare its performance in terms of computational complexity and PSNR performance with Activity based motion estimation scheme

Finally in Chapter 6 we present conclusion and scope for future work followed by references.

Chapter 2 Fundamental Concepts of Motion Estimation

Motion estimation is the computation of the displacement vector between an object in the current frame and that in a stored past frame which is used as the reference. Usually the immediate past frame is used as the reference. Recent video coding standards, such as the H.264 [7] uses as a combination of previously transmitted frame can be used as reference frames thereby offering flexibility in selecting the reference frame.

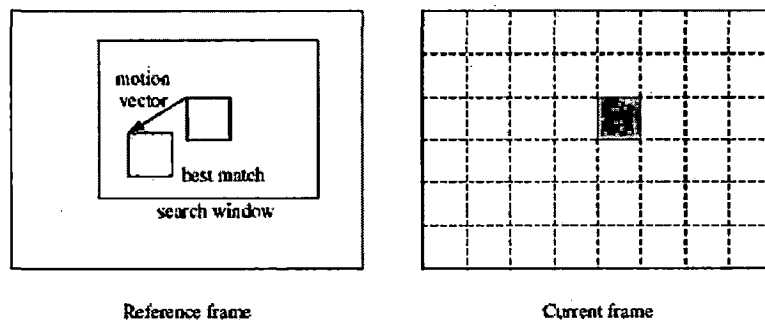


Figure 2-1 General principle of Motion Estimation

Figure 2-1 illustrates the basic philosophy of motion estimation. Consider a pixel belonging to the current frame and then determine its best matching position in the reference frame. The difference in position between the candidate's and its match in the reference frame is defined as the *displacement vector* or the *motion vector*. It is called a vector since it has both horizontal and vertical components of displacement. This algorithm is based on a translational model of the motion of objects between frames. It also assumes that all pixels within the candidate object undergo the same translational movement.

Other approaches to motion estimation use the frequency or wavelet domains etc. New methods of motion estimation are continuously evolved since the process of motion estimation is not specified in coding standards. The standards existing video coding only specify how the motion vectors should be interpreted by the decoder.

2.1 Block Matching Algorithm

It is the most widely used method of motion compensation in which motion is compensated by movement of rectangular sections or 'blocks' of the current frame. An area is searched in the reference frame to find a 'matching' $M \times N$ -sample region. The $M \times N$ block in the current frame is compared with some or all of the possible $M \times N$ regions in the search area and finding the region that give the 'best' match. Matching criterion used is the energy in the

residual block formed by subtracting the candidate region from the current $M \times N$ block so that the candidate region that minimizes the residual energy is chosen as the best match.

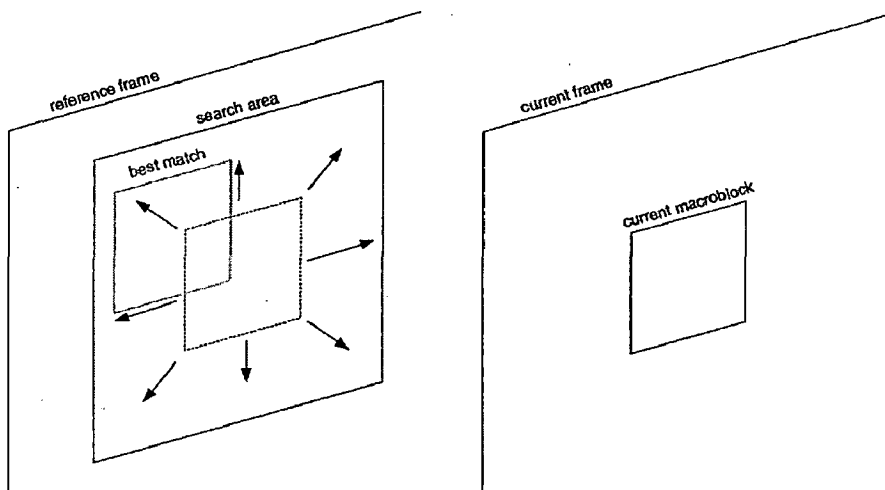


Figure 2-2 Motion Estimation using Block Matching Algorithm

The chosen candidate region becomes the estimate for the current $M \times N$ block and is subtracted from the current block to form a residual $M \times N$ block. The residual block is encoded and transmitted and the offset between the current block (motion vector) and the position of the candidate region is also transmitted.

2.1.1 Backward Motion Estimation

When during motion estimation process the reference frame is the temporally previous frame in the video sequence and the motion of macro-blocks in the previous frame is used to reconstruct the current frame then it is called *backward motion estimation*. Backward motion estimation leads to forward motion prediction.

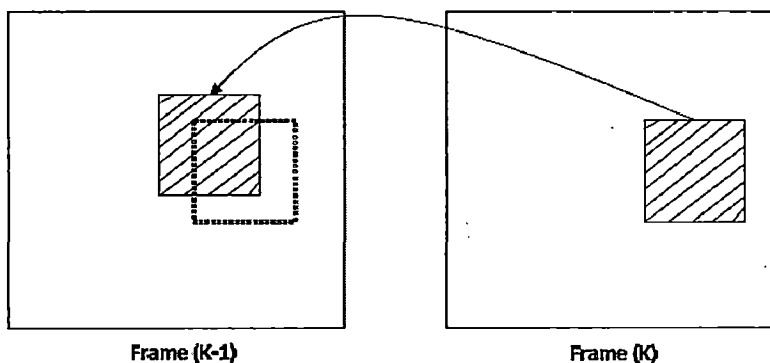


Figure 2-3: Backward Motion estimation with current frame as k and frame $(k-1)$ as the reference frame.

2.1.2 Forward Motion Estimation

It is just the opposite of backward motion estimation. In this case the search for motion vectors is carried out on a frame that appears later than the current frame in temporal ordering. The current frame is estimated from the frame which would be displayed in future. The future frame needs to be transmitted and decoded earlier than the current frame so that a backward motion prediction could be performed. Forward motion estimation leads to backward motion prediction.

Also forward motion estimation leads to a delay in decoding the current frame as a future frame needs to be transmitted and decoded before current frame could be decoded. But forward motion estimation is advantageous in such cases in which some region is covered or uncovered by video object motion which cannot be predicted from previous frames.

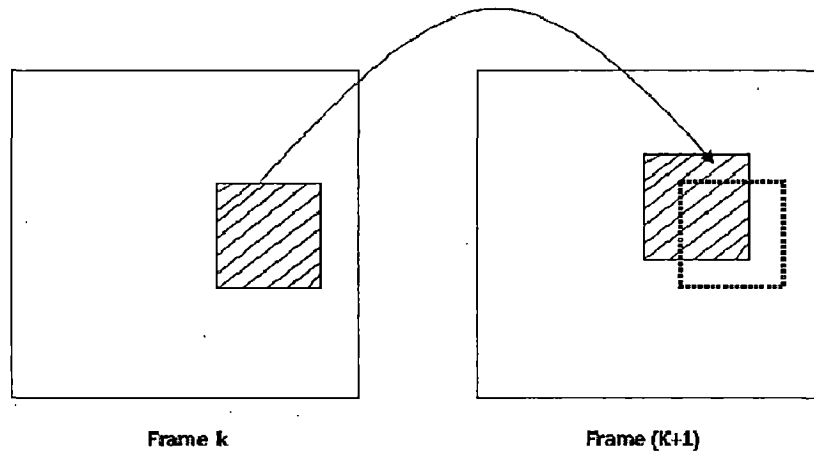


Figure 2-4 Forward Motion estimation with current frame as k and frame (k+1) as the reference frame.

2.1.3 Bi-Directional Motion Estimation

In H.261 [8], only the previous video frame is used as the reference frame for the motion compensated prediction. MPEG-1 allows the future frame to be used as the reference frame for the motion compensated prediction along with the previous frame, which provides better prediction. For example, in figure 2-5 there are moving objects, and if only the forward prediction is used then there will be uncovered areas for which we may not be able to find a good matching block from the previous reference picture (frame N-1). Whereas the backward prediction can properly predict these uncovered areas since they are available in the future reference picture i.e. in frame N + 1.

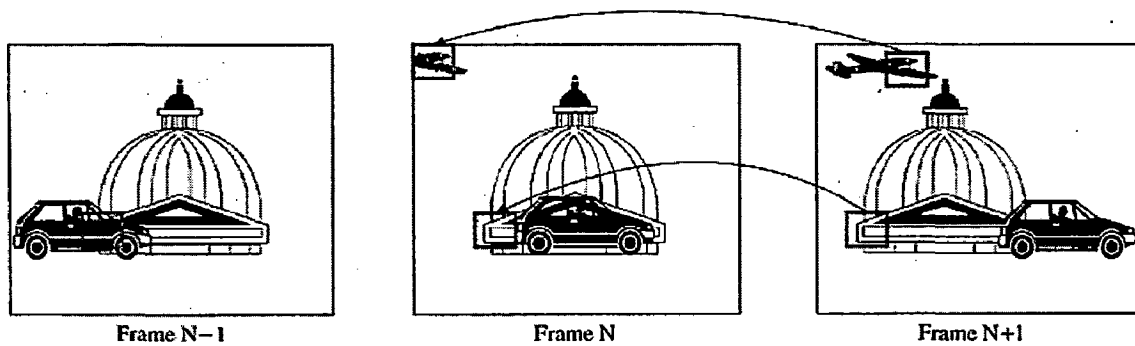


Figure 2-5 A video sequence showing the benefits of bi-directional prediction

In MPEG [9], each video sequence is divided into one or more groups of pictures (GOPs). There are four types of pictures defined in MPEG-1: I, P, B, and D-pictures of which the first three are shown in Fig. 1-6. Each GOP is composed of one or more pictures; one of these pictures must be an I-picture. Usually, the spacing between two anchor frames (I or P-pictures) is referred to as M , and the spacing between two successive I-pictures is referred to as N . In Fig. 1-6, $M = 3$ and

$N = 9$.

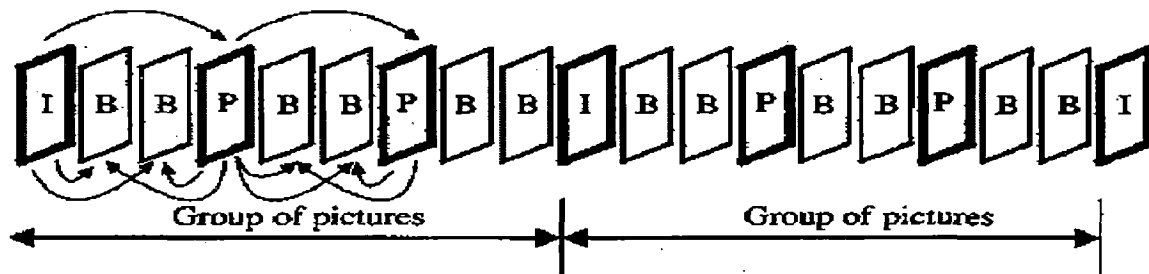


Figure 2-6 MPEG group of pictures.

I-pictures (intra-coded pictures) are coded independently with no reference to other pictures. I-pictures provide random access points in the compressed video data, since the I-pictures can be decoded independently without referencing to other pictures. An MPEG stream with more I-pictures is more editable. Also the error propagation due to transmission errors in previous pictures will be terminated by an I-picture since the I-picture does not reference to the previous pictures. Since I-pictures use only transform coding without motion compensated predictive coding, it provides only moderate compression.

P-pictures (predictive-coded pictures) are coded using the forward motion-compensated prediction from the preceding I or P-picture. P-pictures provide more compression than the I-pictures by virtue of motion compensated prediction. They also serve as references for B-pictures and future P-pictures. Transmission errors in the I-pictures and P-pictures can propagate to the succeeding pictures since the I-pictures and P-pictures are used to predict the succeeding pictures.

B-pictures (bi-directional predicted pictures) allow macro-blocks to be coded using bi-directional motion-compensated prediction from both past and future reference I or P-pictures. In the B-pictures, each bi-directional motion compensated macro-block can have two motion vectors: a forward motion vector which references to a best matching block in the previous I or P-pictures, and a backward motion vector which references to a best matching block in the next I or P-pictures. The effect of noise can be decreased by averaging between the past and the future reference blocks. B-pictures provide the best compression compared to I and P-pictures. I and P-pictures are used as reference pictures for predicting B-pictures. B-pictures are not used as reference pictures hence B-pictures do not propagate errors.

D-pictures (dc-pictures) are low-resolution pictures obtained by decoding only the dc coefficient of the discrete cosine transform (DCT) coefficients of each macro-block. They are not used in combination with I, P, or B-pictures. D-pictures are rarely used, but are defined to allow fast searches on sequential digital storage media.

The trade-off of having frequent B-pictures is that it decreases the correlation between the previous I or P-picture and the next reference P or I-picture. It also causes coding delay and increases the encoder complexity as buffering of frames is required at the decoder.

2.1.4 Block Matching Criteria for Block Matching Algorithm

Inter frame predictive coding is used to eliminate the large amount of temporal and spatial redundancy that exists in video sequences and helps in compressing them. In conventional predictive coding the difference between the current frame and the predicted frame is coded and transmitted. The better the prediction, the smaller the error and hence the transmission bit rate when there is motion in a sequence, then a pixel on the same part of the moving object is a better prediction for the current pixel. There are a number of criteria to evaluate the “goodness” of a match.

Three popular matching criteria used for block-based motion estimation are

1. Mean of squared error (MSE)
2. Sum of absolute difference (SAD)
3. Matching Pel count (MPC)

To implement the block motion estimation, the candidate video frame is partitioned into a set of non overlapping blocks and the motion vector is to be determined for each such candidate block with respect to the reference. For each of these criteria, square block of size $N \times N$ pixels is considered. The intensity value of the pixel at coordinate (n_1, n_2) in the frame k is given by $s(n_1, n_2, k)$ where $(0 \leq n_1, n_2 \leq N-1)$. The frame k is referred to as the candidate frame and the block of pixels defined above is the candidates block.

MSE Criterion

Considering $(k-l)$ as the past references frame $l > 0$ for backward motion estimation, the mean square error of a block of pixels computed at a displacement (i, j) in the reference frame is given by

$$MSE(i, j) = \frac{1}{N^2} \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} [s(n_1, n_2, k) - s(n_1 + i, n_2 + j, k - l)]^2 \quad (2.1)$$

Consider a block of pixels of size $N \times N$ in the reference frame, at a displacement of, where i and j are integers with respect to the candidate block position. The MSE is computed for each displacement position (i, j) , within a specified search range in the reference image and the displacement that gives the minimum value of MSE is the displacement vector which is more commonly known as motion vector and is given by

$$[d_1, d_2] = \underset{i, j}{\operatorname{argmin}} [MSE(i, j)] \quad (2.2)$$

The MSE criterion defined in equation 2.1 requires computation of N^2 subtractions, N^2 multiplications (squaring) and $(N^2 - 1)$ additions for each candidate block at each search position. This is computationally costly and a simpler matching criterion, as defined below is often preferred over the MSE criterion.

SAD Criterion

Like the MSE criterion, the sum of absolute difference (SAD) too makes the error values as positive, but instead of summing up the squared differences, the absolute differences are summed up. The SAD measure at displacement (i, j) is defined as

$$SAD(i, j) = \frac{1}{N^2} \sum_{n_1}^{N-1} \sum_{n_2}^{N-1} [s(n_1, n_2, k) - s(n_1 + i, n_2 + j, k - l)] \quad (2.3)$$

The motion vector is determined in a manner similar to that for MSE as

$$[d_1, d_2] = \underset{i, j}{\operatorname{argmin}}[SAD(i, j)] \quad (2.4)$$

The SAD criterion shown in equation 2.3 requires N^2 computations of subtractions with absolute values and additions N^2 for each candidate block at each search position. The absence of multiplications makes this criterion computationally more attractive and facilitates easier hardware implementation.

MPC Criterion

In this criterion, the pixels of the candidates block B are compared with the corresponding pixels in the block with displacement (i, j) , in the reference frame and those which are less than a specified threshold, i.e., closely matched are counted. The count for matching and the displacement (i, j) , for which the count is maximum correspond to the motion vector. We define a binary valued function as

$$count(n_1, n_2) = \begin{cases} 1 & \text{if } |s(n_1, n_2, k) - s(n_1 + i, n_2 + j, k - l)| \leq \theta \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

where, θ is a pre-determined threshold. The matching pel count (MPC) at displacement (i, j) is defined as the accumulated value of matched pixels as given by

$$MPC(i, j) = \sum_{n_1}^{N-1} \sum_{n_2}^{N-1} [count(n_1, n_2)] \quad (2.6)$$

$$[d_1, d_2] = \underset{i, j}{\operatorname{argmin}}[MPC(i, j)] \quad (2.7)$$

2.1.5 Block Size

The energy in the residual is reduced by motion compensating each 16×16 macro-block. Motion compensating each 8×8 block (instead of each 16×16 macro-block) reduces the residual energy further and motion compensating each 4×4 block gives the smallest residual

energy of all. Smaller motion compensation block sizes can produce better motion compensation results.

However, a smaller block size leads to increased complexity (more search operations must be carried out) and an increase in the number of motion vectors that need to be transmitted. Sending each motion vector requires bits to be sent and the extra overhead for vectors may outweigh the benefit of reduced residual energy. An effective compromise is to adapt the block size to the picture characteristics, for example choosing a large block size in flat, homogeneous regions of a frame and choosing a small block size around areas of high detail and complex motion. H.264 uses an adaptive motion compensation block size [7].

2.1.6 Search Range

The maximum allowed motion displacement d_m , also known as the search range, has a direct impact on both the computational complexity and the prediction quality of the BMA. A small d_m results in poor compensation for fast-moving areas and consequently poor prediction quality. A large d_m , on the other hand, results in better prediction quality but leads to an increase in the computational complexity (since there are $(2d_m+1)^2$ possible blocks to be matched in the search window). A larger d_m can also result in longer motion vectors and consequently a slight increase in motion overhead [10]. In general, a maximum allowed displacement of $d_m = \pm 15$ pixels is sufficient for low-bit-rate applications. MPEG standard uses a maximum displacement of about ± 15 pixels, although this range can optionally be doubled with the unrestricted motion vector mode.

2.1.7 Search Accuracy

In some cases a better motion compensated prediction may be formed by predicting from interpolated sample positions in the reference frame. The reference region luma samples are interpolated to half-sample positions and it may be possible to find a better match for the current macro-block by searching the interpolated samples. 'Sub-pixel' motion estimation and compensation involves searching sub-sample interpolated positions as well as integer-sample positions, choosing the position that gives the best match (i.e. minimises the residual energy) and using the integer or sub-sample values at this position for motion compensated prediction. This approach may be extended further by interpolation onto a quarter-sample grid to give a still smaller residual. In general, 'finer' interpolation provides better motion compensation performance (a smaller residual) at the expense of increased complexity. The performance gain tends to diminish as the interpolation steps increase. Half-sample

interpolation gives a significant gain over integer-sample motion compensation, quarter-sample interpolation gives a moderate further improvement, eighth-sample interpolation gives a small further improvement again and so on. Searching for matching 4×4 blocks with quarter-sample interpolation is considerably more complex than searching for 16×16 blocks with no interpolation. In addition to the extra complexity, there is a coding penalty since the vector for every block must be encoded and transmitted to the receiver in order to reconstruct the image correctly. As the block size is reduced, the number of vectors that have to be transmitted increases. More bits are required to represent half- or quarter-sample vectors because the fractional part of the vector (e.g. 0.25, 0.5) must be encoded as well as the integer part. There is therefore a trade-off in compression efficiency associated with more complex motion compensation schemes, since more accurate motion compensation requires more bits to encode the vector field but fewer bits to encode the residual whereas less accurate motion compensation requires fewer bits for the vector field but more bits for the residual.

2.2 Fast Block Matching Motion Estimation Algorithms

Motion compensation aims to minimize the energy of the residual transform coefficients after quantization which usually involves evaluating the residual energy at a large number of different offsets which is computationally extensive. The metric used as a measure of residual energy is called Block Distortion Measure (BDM). Sum of Absolute Difference (SAD) is mostly used as BDM during block matching.

Full Search motion estimation involves evaluating SAD at each point in the search window. Full search searches all search locations in search window so it is more computation extensive but always gives best possible match. In computation- or power-limited applications fast search algorithms are preferable. These algorithms operate by calculating the energy measure (e.g. SAD) at a subset of locations within the search window. Some of them are listed below:

2.2.1 Three Step Search (TSS)

As shown in Figure 2-7, the three step search starts the search by evaluating SAD at the corners, middle of edges and centre of search window. There are a total of nine points. Then the search window (usually starting size is 9) of half the size is centered at the position of minima from previous search window [11]. This process is repeated till the size of search window couldn't be reduced further. This is also called N -step search as it involves N steps of search window size for a given search window with $2^N - 1$ samples.

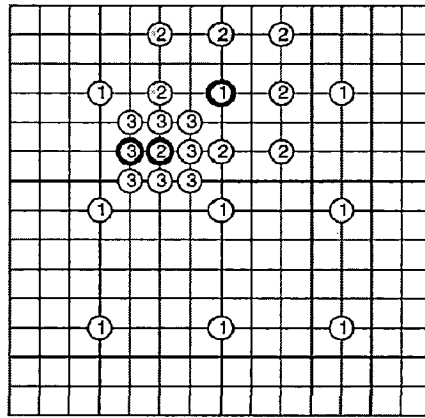


Figure 2-7 Three step search

The TSS is considerably simpler than Full Search ($8N + 1$ searches compared with $(2^{N+1} - 1)^2$ searches for Full Search) but the TSS does not usually perform as well as full search in terms of SAD.

2.2.2 Four Step Search (4SS)

For the maximum motion displacements of ± 7 , the 4SS algorithm utilizes a center-biased search pattern with nine checking points on a 5×5 window in the first step instead of a 9×9 window in the 3SS. The center of the search window is then shifted to the point with minimum SAD. If the minimum BDM point is found at the center of the search window, the search will go to the final step i.e. the search window is reduced to 3×3 and the search stops at this small search window. Otherwise, the search window size is maintained at 5×5 size and two cases are possible. First, if the previous minimum BDM point is located at the corner of the previous search window, five additional checking points as shown in Figure 2-8 (b) are used. Second, if the previous minimum BDM point is located at the middle of horizontal or vertical axis of the previous search window, three additional checking points as shown in Figure 2-8(c) are used. If the minimum BDM point is found at the center of the search window then go to final step otherwise repeat the above with same 5×5 search size once and then go to final step.

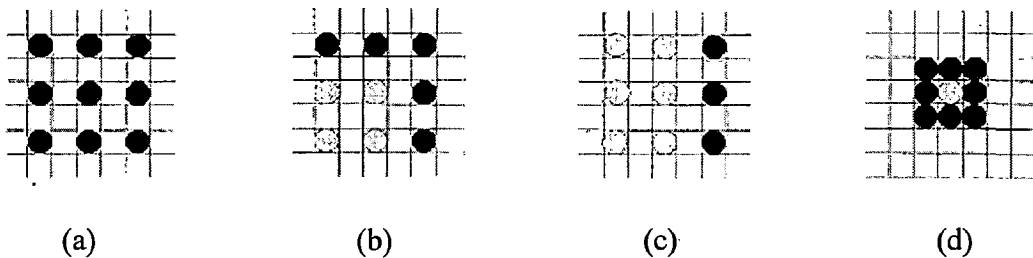


Figure 2-8 Search patterns of the 4SS. (a) First step, (b) intermediate step, (c) alternate step, and (d) last step.

Experimental results have shown that the proposed 4SS algorithm performs better [12] than the well-known 3SS. Also 4SS is more robust as compared with 3SS because the performance of 4SS is maintained for image sequence that contains complex movement such as camera zooming and fast motion.

2.2.3 Diamond Search (DS)

The proposed DS algorithm employs two search patterns as shown in Figure 2-9. The larger pattern comprising of nine checking points from which eight points surround the center one to compose a diamond shape is called large diamond search pattern (LDSP). The smaller pattern consisting of five checking points forms a smaller diamond shape called small diamond search pattern (SDSP).

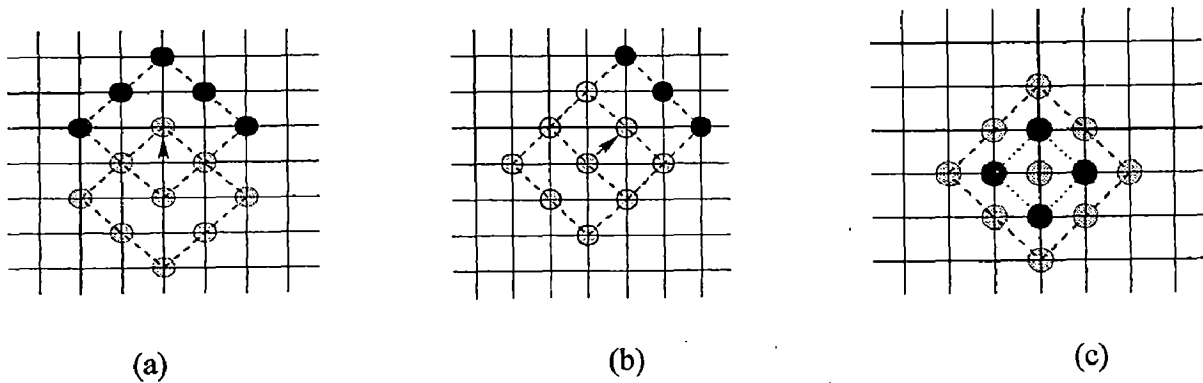


Figure 2-9 Diamond Search algorithm, (a) The corner point (LDSP->LDSP) (b) The edge point (LDSP->LDSP)(c) The centre point(LDSP->SDSP)

In diamond search the search begins with LDSP search pattern centred at the position of current macro block. If the position corresponding to minimum block distortion (MBD) is at the corner of LDSP then the search is repeated with the new LDSP centred at that macro-block. If the MBD position corresponds to the edge then the whole LDSP is shifted in that direction so that the centre of new LDSP occupies that position. If the position of MBD corresponds to the centre position, the LDSP is replaced by a SDSP for the evaluation of block distortion measure. The MBD point found in this step is the final solution of the motion vector which points to the best matching block.

DS algorithm significantly outperforms the well-known TSS algorithm [13] in terms of computations. Compared with 4SS DS algorithm also works better on average in terms of MSE values, reconstructed image quality, and average number of search points. The combinations of DS and Hexagon Based Search Pattern have been proposed which provides better performance. [14]

2.2.4 Comparison of Fast Block Matching Motion Estimation Algorithms

Various fast block matching motion estimation algorithms discussed in the previous section are compared in figure 2-10 and figure 2-11 based on the PSNR performance and number of computations of BDM per macro-block respectively. Here current frame is estimated from the previous frame as the reference frame. Search parameter is ± 8 .

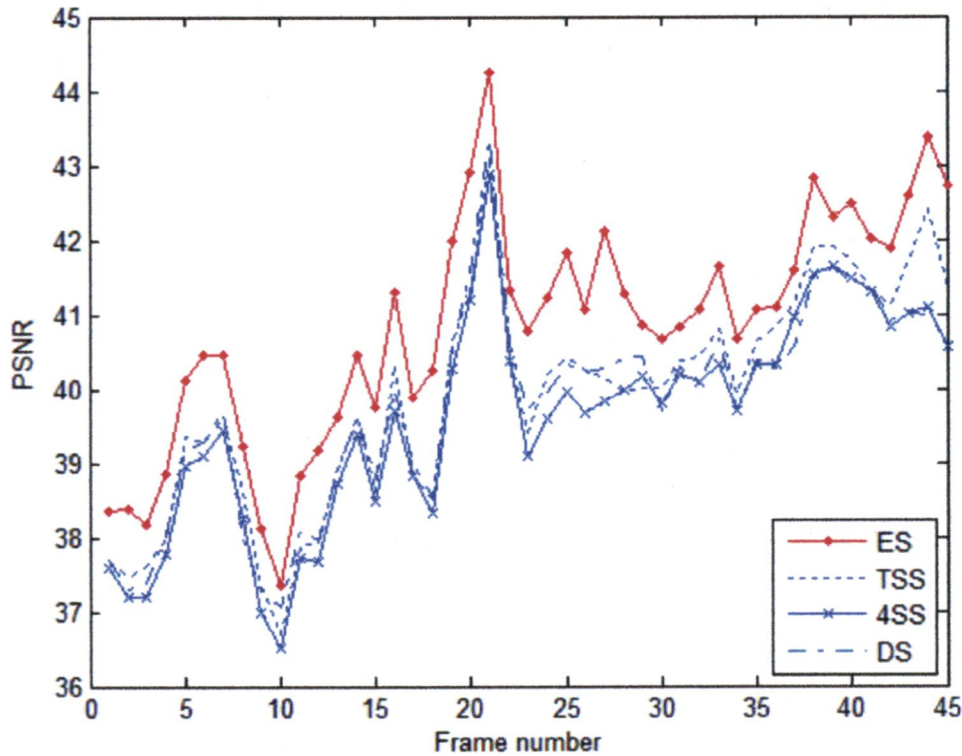


Figure 2-10 Comparative PSNR performance for ICE video sequence, at QCIF resolution, 15 fps for ES, TSS, 4SS and DS

From figure 2-10 we can see that all the fast search algorithms show almost similar PSNR performance which is significantly worse than that of full search.

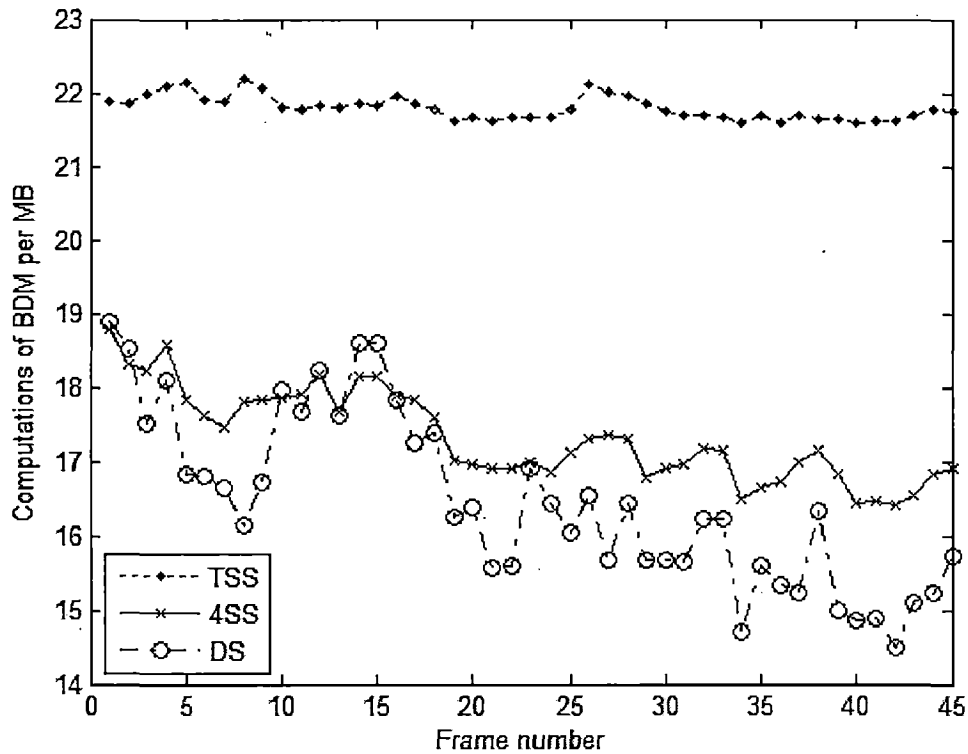


Figure 2-11 Comparison of Number of BDM computations per MB for ICE video sequence, at QCIF resolution, 15 fps for TSS, 4SS and DS

From figure 2-11 we observe that DS is least computationally expensive among other fast search algorithms and provide similar PSNR performance as other fast search algorithms.

All those applications where better compression efficiency is required because of limited bandwidth exhaustive search is preferred as it results in less residual error which is to be encoded separately. For those applications where computational power is limited fast search algorithms are used at the cost of reduced compression efficiency.

Chapter 3 Motion Estimation with Sub-Sample Accuracy (SSA)

Video compression algorithms use block based motion estimation and compensation in order to reduce temporal redundancy for achieving compression. Although whole pixel motion estimation provides a high degree of redundancy reduction, it is found that sub-pixel accurate motion estimation further reduces temporal redundancy. Half pixel and Quarter pixel accurate motion estimation are generally used in which macro-blocks at half pixel and quarter pixel positions respectively are also searched along with macro-blocks at integer pixel positions. Such macro-block search at fractional pixel position accounts for half-pixel and quarter pixel motion of objects in natural video sequences.

Sub-pixel motion estimation and compensation involves interpolation of pixel positions at sub-pixel positions and choosing the position that gives the best match i.e. minimizing the residual energy and using the integer- or sub-sample values at this position for motion compensated prediction. Interpolation of whole image at sub-pixel positions is not only time consuming but also increases the number of macro-blocks to be searched by a factor of 4 and 16 for half-pixel and quarter pixel accuracy respectively, hence making sub-sample accurate motion estimation a more computationally intensive exercise. This is evident from the comparison of CPU time and number of computations of Block Distortion Measure per Macro-Block (Δ CBDM/MB) for Motion Estimation at integer pixel and half pixel accuracy in Table 3-1.

3.1 Conventional Sub-Sample Accurate Motion Estimation (SSA ME)

Sub-pixel accuracy for motion estimation is traditionally made possible using interpolated reference frames [7]. The creation and use of such interpolated reference frames has a significant implication on computational load of the encoder whereas the PSNR performance of motion estimation improves significantly with the use of sub-pixel accurate motion vectors. Figure 3-1 shows the comparison of PSNR of motion compensated frames using integer-pixel accurate motion vectors with that using half-pixel accurate motion vectors.

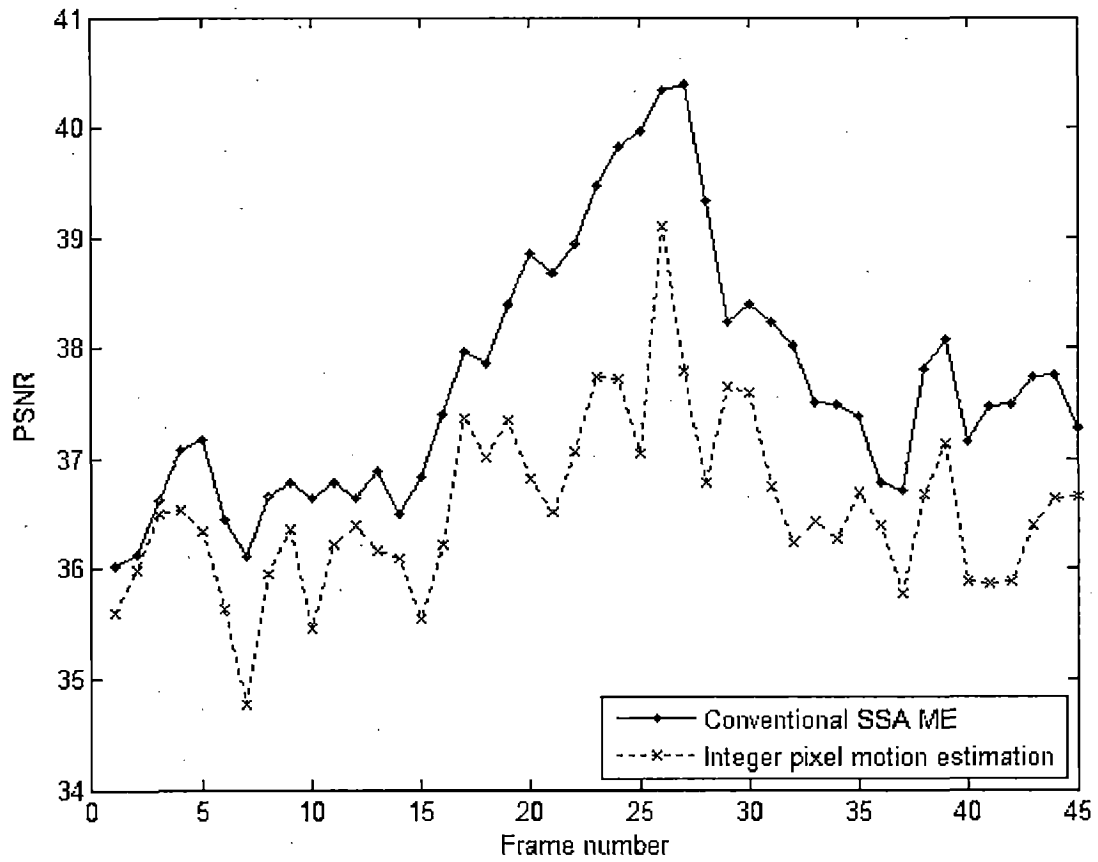


Figure 3-1: Plot of PSNR for Integer pixel Accuracy Motion estimation and Compensation Vs PSNR for Conventional Half pixel Accurate Motion estimation and compensation. Video sequence: Soccer, 15 fps, Format: QCIF, Macro-block Size: 16.

Next we present a detailed analysis and comparison of conventional Half-Pixel Accurate Motion Estimation (HPA ME) with Integer-Pixel Accurate Motion Estimation (IPA ME) for a set of five standard video sequences at QCIF resolution and 15fps frame rate. The CPU times, number of SAD computations per macro-block and PSNR are stated as a mean of their values for first 45 frames of respective video sequences.

To analyze only the effect of search algorithm, unlike motion estimation in video coding applications, backward motion estimation is performed between successive original frames and PSNR is calculated between original and its corresponding motion compensated frame. This approach avoids the influence of rate-distortion optimization and error propagation. Motion Estimation is performed by using only luminance component, and sum of absolute difference as the block distortion measure.

Video Sequence	Conventional HPA PSNR(dB)	IPA PSNR(dB)	Δ PSNR(dB)	Δ CBDM/MB (%)	Δ T (%)
<i>Soccer</i>	37.6967	36.5563	1.1404	274.42	282.18
<i>City</i>	36.7847	35.0700	1.7146	274.42	280.05
<i>Crew</i>	37.3934	36.6851	1.3958	274.42	278.38
<i>Harbour</i>	33.3677	31.9719	1.3654	274.42	274.77
<i>Ice</i>	40.7696	39.7604	1.0092	274.42	277.60
<i>Average</i>	-	-	1.3251	274.42	278.60

Table 3-1: Comparison of Conventional Half-Pixel Accurate (HPA) motion estimation with Integer-Pixel Accurate (IPA) motion estimation based on PSNR, average increase in CPU time (Δ T) and increase in number of computations of Block Distortion Measure per Macro-Block (Δ CBDM/MB).

In Table 3-1 we observe that there is a gain in PSNR performance of Motion estimation by 1.33 dB at the cost of 278% increase in CPU time. Hence, some fast algorithms are required to speed up the sub-pixel accurate motion estimation process.

Fast sub-pixel accurate motion estimation techniques have been proposed in [15] and [16] which decrease the computational load on the encoder significantly by fitting a parabolic surface to estimate Block Distortion Measure(BDM) at sub-pixel positions by using BDM values at the proximity of macro-block given by integer pixel accurate motion vector. The detailed description of these algorithms is provided in Section 3.2.

3.2 Interpolation Free Sub-pixel Accurate Motion Estimation

Here we introduce an interpolation-free method for sub-pixel block based motion estimation which reduces memory bandwidth requirements and improves computational efficiency as proposed by P. R Hill, T. K. Chiew, D. R. Bull and C.N. Canagarajah (Interpolation Free Subpixel Accuracy Motion Estimation, 2006) [15]

In this method a parabolic model of sub-pixel resolution motion estimation is described that uses the sum of absolute difference (SAD) cost at the best whole pixel resolution position and its neighbours for estimating the SAD cost at sub-pixel positions. In Section 3.2.1 this parabolic model is defined and Section 3.2.2 describes the generation of the parameters for the model described in Section 3.2.1.

3.2.1 Model Description

The motion vector is estimated in a coarse to fine fashion. The coarse estimate of motion vector is obtained using integer pixel motion estimation using standard block matching algorithms. For fine estimation a parametrically controlled parabolic surface is used to estimate the sub-pixel SAD values approximating the actual behaviour of SAD in case of pure translations. This parabolic surface is given by Equation 3.1

$$SAD_i(x,y) = Ax^2 + By^2 + Cxy + Dx + Ey + F \quad 3.1$$

Here, $SAD_i(x,y)$ is the estimated SAD value of the i th macro-block. The x and y are the coordinates of estimation centred at the best motion vector at integer pixel resolution. The values of x and y may vary from -1.0 to +1.0. The coordinate (0, 0) is the location of best motion vector at integer pixel accuracy. Therefore, for half-pixel accuracy x and y can take values $[-1, -\frac{1}{2}, 0, \frac{1}{2}, +1]$ as used in simulation of implementation of this technique. After integer pixel accurate motion vector is obtained, the SAD values at its nearest neighbours are obtained giving eight nearest SAD neighbours from which A, B, C, D, E and F can be estimated. These eight neighbours are shown in Figure 3.2.

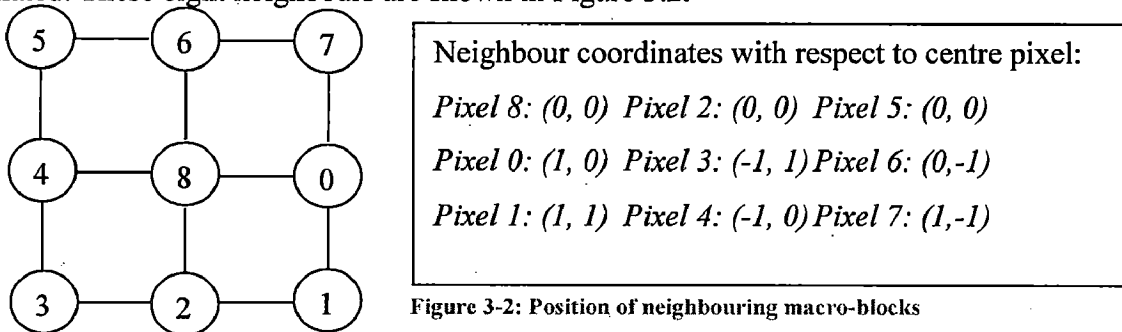


Figure 3-2: Position of neighbouring macro-blocks

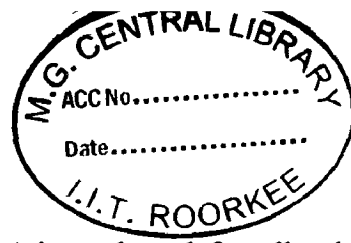
There are nine potential points for the estimation of the six parameters in equation 3.1 therefore the system is over-described there are therefore many possible estimation methods. These methods include an under-determined model based on just near neighbours (with even indices) defined as near neighbour model (NNM), an over-complete system model (OSM) that used all 9 neighbours and a complete system model (CSM). According to Chiew and Hill et al [2] (Interpolation Free Subpixel Accuracy Motion Estimation, 2006) CSM is proved to give better results and hence used for the implementation of this technique.

3.2.2 Parameter estimation

In CSI model, the parameters A, B, D, E and F are calculated from equation 3.2 and 3.3. Equation 3.2 is derived by substitution of neighbours coordinates and computed SAD value corresponding to that macro-block in equation 3.1. However, the value of C is chosen from $C_1, C_3, C_5,$ and C_7 where C_k is the value of C found with the complete system of equations using points 0, 2, 4, 6, 8 and k as defined in figure 3-2. The value of C is chosen from equation 3.4 where \hat{S}_{ki} is the estimate of S_i using equation 3.1 with parameter set A, B, C_k , D, E, F. This model determines which of the far-neighbours best fits the model and ignores the other 3.

$$\begin{aligned}
S_0 &= S(1,0) = A + D + F \\
S_1 &= S(1,1) = A + B + C + D + E + F \\
S_2 &= S(-1,1) = B + E + F \\
S_3 &= S(-1,0) = A + B - C - D + E + F \\
S_4 &= S(-1,-1) = A - D + F \\
S_5 &= S(0,-1) = A + B + C - D + E + F \\
S_6 &= S(1,-1) = B - E + F \\
S_7 &= S(0,0) = A + B - C + D - E + F \\
S_8 &= S(1,0) = F
\end{aligned} \tag{3.2}$$

$$\begin{aligned}
A &= -S_8 + \frac{1}{2}(S_0 + S_4) \\
B &= -S_8 + \frac{1}{2}(S_2 + S_6) \\
D &= \frac{1}{2}(S_0 - S_4) \\
E &= \frac{1}{2}(S_0 - S_6) \\
F &= S_8
\end{aligned} \tag{3.3}$$



$$C = \operatorname{argmin}_{k=1,3,5,7} \sum_{i=1,3,5,7} |S_i - \hat{S}_{ki}|$$

3.4

To obtain the parabolic surface minimum $S(x, y)$ is evaluated for all sub-pixel locations within the $(-1,1) \times (-1,1)$ area. The thus obtained coordinate vector is added to the coarse motion vector obtained by integer pixel accurate motion estimation to get the sub-pixel accurate motion vector.

3.2.3 Results of Interpolation-Free Sub-pixel Accurate Motion Estimation

To analyze only the effect of search algorithm, unlike motion estimation in video coding applications, backward motion estimation is performed between successive original frames and PSNR is calculated between original and its corresponding motion compensated frame. This approach avoids the influence of rate-distortion optimization and error propagation. Motion Estimation is performed by using only luminance component, and Sum of Absolute Difference (SAD) as the block distortion measure.

Exhaustive search is used first for obtaining coarse estimate of motion vector for a frame and then motion vector refinement is performed using the mentioned interpolation-free technique. This sub-pixel accurate motion technique is performed for five standard video sequences and the results are tabulated in Table 3-2.

The CPU times, number of SAD computations per macro-block and PSNR are stated as a mean of their values for first 45 frames of respective video sequences.

This interpolation free method is inferior to the conventional interpolation method in terms of the PSNR gain. This is because the parabolic surface which is used to estimate SAD at sub-pixel macro-block locations gives good estimate of SAD in case of pure translations only [15] which may not be the case in real world video sequences as motions like pan, zoom, rotation etc also take place. Hence, the sub-pixel accurate motion vector thus obtained by interpolation free technique is a suboptimal estimate of sub-pixel accurate motion vector obtained by conventional sub-pixel accurate motion estimation using interpolation.

However, the sub-pixel accurate motion estimation and compensation using this method provides a better PSNR over that of integer pixel accurate motion estimation and compensation. Figure 3-3 compares PSNR obtained through this interpolation-free SSA ME method to that obtained by Integer-pixel accurate motion estimation (IPA ME).

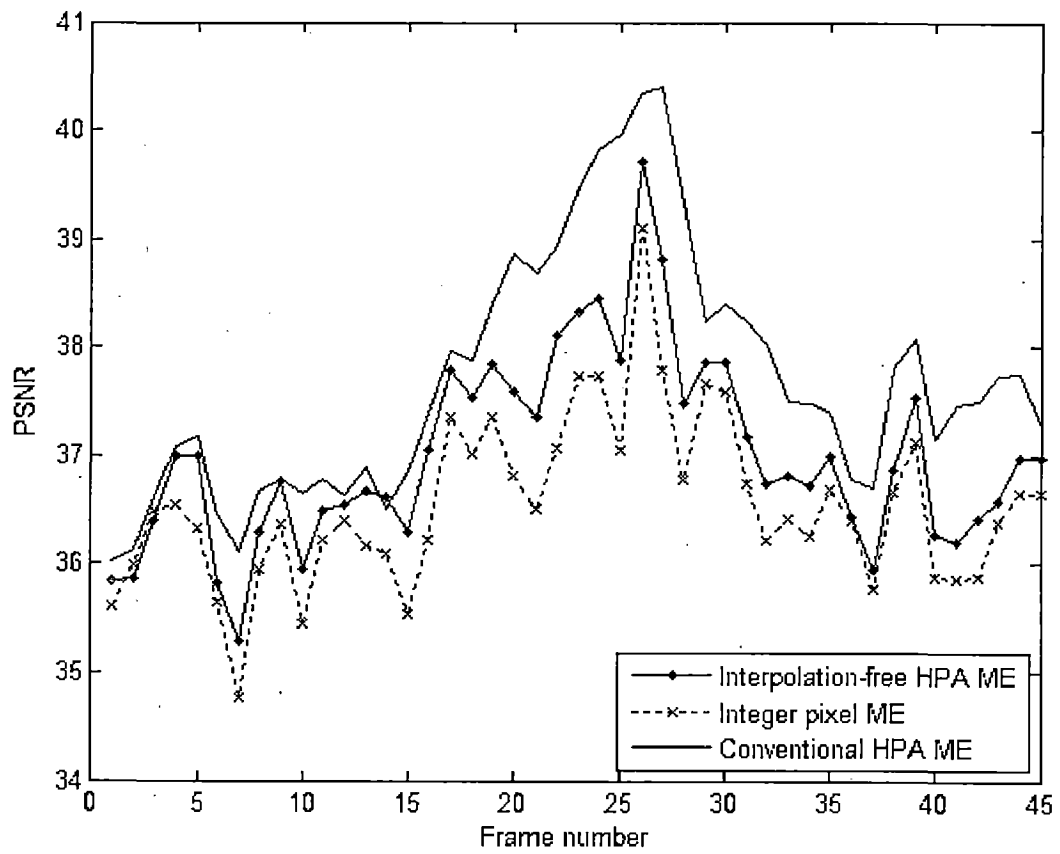


Figure 3-3: Comparison of Interpolation-free Half-Pixel Accurate Motion Estimation (HPA ME), Conventional half-pixel Motion Estimation and Integer pixel Motion Estimation. : Soccer, Format: QCIF, Macro-block Size: 16, search parameter: 8.

Also there is only a marginal increase in number of SAD computations (CBDM/MB) per macro-block and CPU time (T) as compared to integer pixel accurate motion estimation. The CPU times, number of SAD computations per macro-block and PSNR are recorded for various video sequences using this interpolation-free method in Table 3-2.

Video Sequence	Interpolation free HPA ME PSNR(dB)	IPA ME PSNR(dB)	Δ PSNR(dB)	Δ CBDM/MB (%)	Δ T (%)
<i>Soccer</i>	37.0036	36.5563	0.4473	2.8412	3.3787
<i>City</i>	35.3184	35.0700	0.2484	3.0196	3.6086
<i>Crew</i>	36.8262	36.6851	0.1411	2.6405	3.6448
<i>Harbour</i>	32.1200	31.9719	0.1481	2.6004	3.2628
<i>Ice</i>	40.1074	39.7604	0.3470	2.6918	3.8063
<i>Average</i>			0.2664	2.7587	3.5402

Table 3-2: Comparison of Interpolation-free Half-Pixel Accurate (HPA) motion estimation with Integer-Pixel Accurate (IPA) motion estimation based on PSNR, average increase in CPU time (Δ T) and increase in number of computations of Block Distortion Measure per Macro-Block (Δ CBDM/MB).

This technique acts as an efficient trade off between improvement in PSNR due to better prediction and computation time consumed by motion estimation. Here, PSNR gain of 0.27dB is achieved with 3.54% increase in CPU time required for motion estimation.

Chapter 4 Motion Estimation for Scalable Video Coding (SVC)

4.1 Introduction to Scalable Video Coding (SVC)

The resolution diversity of current display devices motivates the need of spatial scalability as transmitting a single representation of video sequence to a range of display resolutions available in the market is impractical. A device with low display resolution with the capacity for decoding and down-sampling high resolution material is not justified as such a requirement could increase cost and power of the device. Spatial scalability allows the compression of multiple video sequences with same content but different resolution. To remove redundancy between neighbouring layers, spatial scalability exploits the inter-layer motion prediction. In SVC with multiple layers having different resolutions, reducing redundancy among ME processes in different layers is critical to reduce the overall time complexity.

Various fast ME approaches based on the dynamic search range adjustment have been proposed to reduce the computational complexity. In [17], the search range is determined according to the magnitude of prediction errors. Yamada et al. [18] also proposed an adaptive search range selection algorithm based on the sum of the absolutes of motion vectors and prediction errors in the previous frame. Song et al. [19] utilized the average motion vectors in five previous reference frames and the prediction error of the current block simultaneously. In [20], the motion vector difference is utilized to predict the search range.

Many algorithms have been proposed to reduce computational time in ME by employing hierarchical search with search pattern. Three step search (TSS) [11], 4-step search (4SS) [12] and diamond search (DS) [13] are centre biased search algorithms using specific search pattern such as rectangle or diamond. These algorithms are faster than others but also result in significant decrement in PSNR. Moreover in case of SVC these algorithms do not take advantage of motion information from base layer for encoding of enhancement layers.

Sangwon Na et al. utilized a new activity based motion estimation scheme [21] to reduce the computational complexity of multilayer motion estimation for scalable video coding. This approach is based on the activity which is defined as the absolute difference between the motion vector predictor and the final motion vector. In Section 4.2 the implementation and results obtained with this activity based approach are discussed for motion estimation of

video sequence at three different spatial resolutions. The simulations are performed to determine the PSNR performance and computational efficiency of the method.

4.2 Activity-Based Motion Estimation Scheme for Scalable Video Coding

This method is based on the correlation of the activities between neighbouring layers. An inter-layer activity model using a curve-fitted linear equation is applied to exploit the activity in the base layer for deciding the search centre and the search range of the enhancement layer. Each activity pair in the neighbouring layers is used to associate the relevant macro-block in enhancement layer to one of two groups: boundary region MB and interior region MB. Based on the type of MB i.e. inner or outer, a minimal sufficient search range is decided from the inter-layer activity prediction factor that is adjusted to the given sequence.

In SVC the motion vector (MV) and the motion vector difference (MVD) of the lower resolution layer, referred to as base layer, is available for the encoding of the enhancement layer. With the assumption that MVs in neighbouring layers are correlated, the bit-rate of SVC can be significantly reduced by utilizing the BL motion vector predictor (MVP). BL motion vector predictor is MV obtained by up-scaling the MV of the base layer (BL) corresponding macro-block. BL MVP also helps significantly reducing search range since BL MVP is quite close to the best motion vector in terms of the SAD cost. Inter-layer activity model is to decide the search range for achieving high PSNR performance. Parameters used in this model are adjusted to the given sequence automatically.

4.2.1 Spatial Scalability

Spatial scalability for multilayer coding provides multiple resolutions with a single coded bit stream. The ratio of vertical (or horizontal) dimensions between the layers measured as Scale factor. In the dyadic case the vertical and horizontal dimensions grow by a scale factor of 2 between neighbouring layers. This scale factor is used to calculate the corresponding pixel positions between two neighbouring layers.

To improve the coding efficiency, inter layer prediction schemes are adopted. Inter-layer prediction scheme use signals of the base layer to predict those in the enhancement layer thereby improving the rate-distortion performance. There are three inter-layer prediction schemes: inter-layer intra texture prediction, inter-layer motion prediction and inter-layer

residual prediction. In this activity based motion estimation technique only Inter-Layer Motion Prediction (ILMP) is used. In ILMP the motion vectors are derived from the base layer. For example, 8×8 MB in base layer corresponds to 16×16 MB in enhancement layer and motion vectors of base layer are scaled by a factor of 2, which are used as motion vector predictors for a new macro block in enhancement layer i.e. BL MVP.

4.2.2 Base Layer Motion Vector Predictor (BL MVP)

Based on the correlation of motion vectors in the neighbouring Layers, the approach adopted has BL MVP for enhancement layer and MV of BL as depicted in figure 4-1

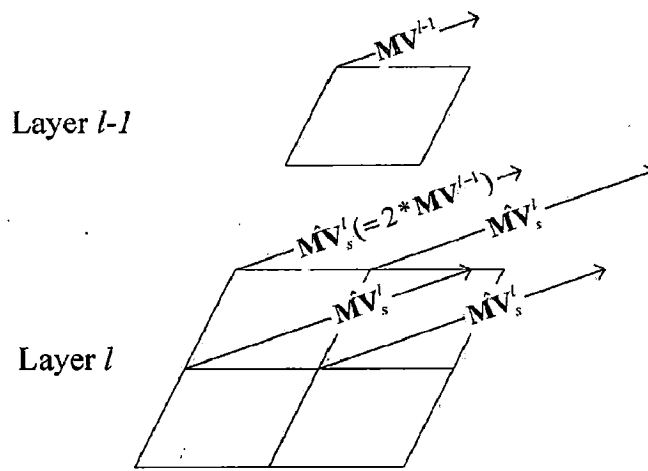


Figure 4-1 MV^{l-1} , which denotes the final MV at the base layer (layer $l-1$), is used to obtain \hat{MV}_s^l , i.e., the base-layer motion vector predictor of the corresponding MBs at the enhancement layer (layer l).

To validate the model being used, the distribution of difference of BL MVP and the final MV within the minimum bounding box (MBB) which covers both as shown in figure 4-2 is evaluated. L_{MBB} denotes the length of longer edge of the MBB.

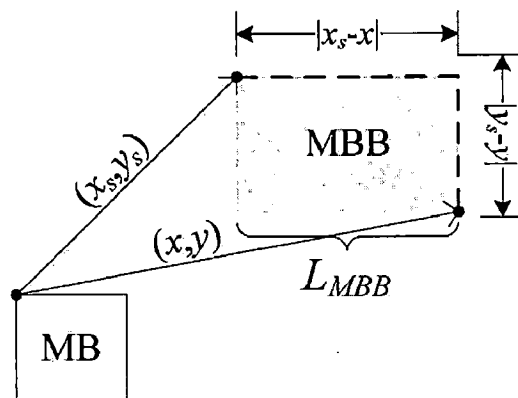


Figure 4-2 Length of the longer edge of the MBB, L_{MBB} where (x_s, y_s) denotes the BL MVP, and (x, y) denotes the final motion vector.

To analyze only the effect of search algorithm, backward motion estimation is performed between successive original frames and PSNR is calculated between original and its corresponding motion compensated frame. Full Search is used at both the QCIF and CIF resolutions. Motion Estimation is performed by using only luminance component, and sum of absolute difference as the block distortion measure. This approach avoids the influence of rate-distortion optimization and error propagation.

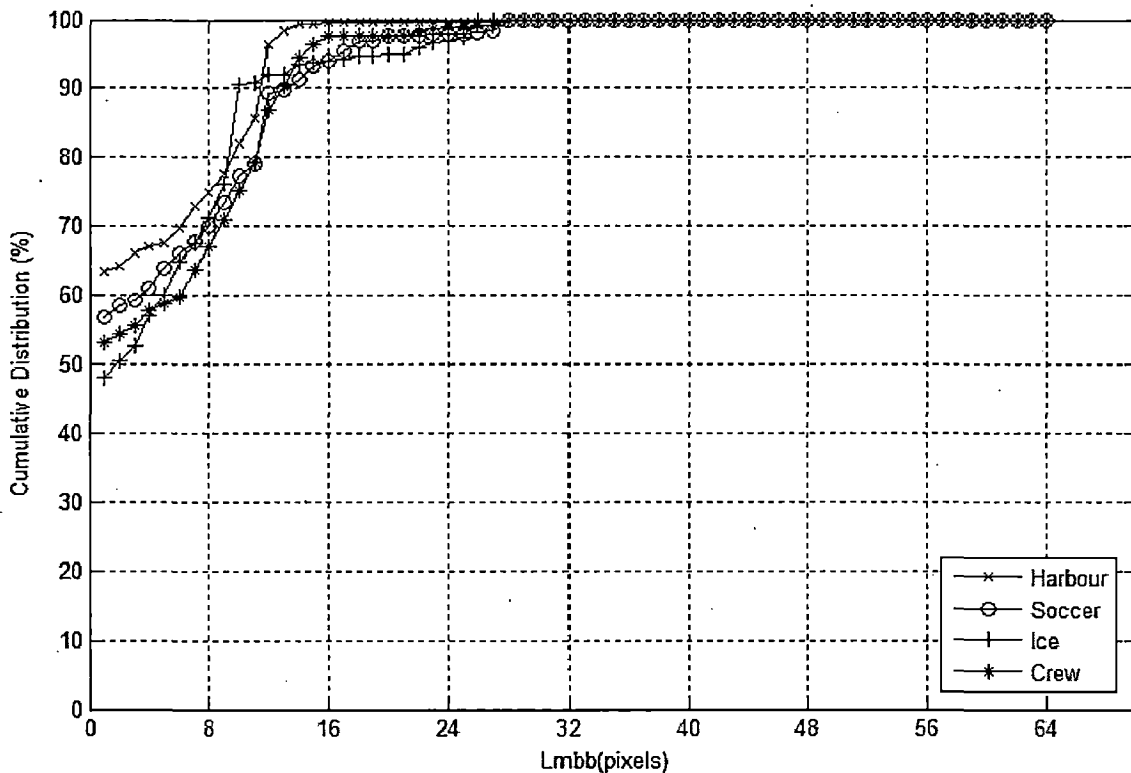


Figure 4-3 Cumulative distribution of L_{MBB} , the maximum length of MBB, for video sequences: Harbour, Soccer, Ice and Crew. QCIF and CIF as base layer and enhancement layer respectively

From figure 4-3 it is observed that majority of MV's of the enhancement layer can be found within $[-8,+8]$ of search centre at $\hat{M}V_s$. Hence basis search range, SR_{basis} is set at 8.

The cumulative distribution as presented here is somewhat different from that reported in [21] because here only a single frame is used as a reference frame whereas in [21] hierarchical prediction structures are used for motion compensated prediction with $GOP=8(IBBBBBBBP)$.

Besides the BL MVP, there are some other efficient predictors [24]. Conventional median predictor is usually employed in recent video compression. To minimize the memory bandwidth and retain the processing regularity in hardware implementation, many very large

scale integration video coders adopt zero motion vector $(0, 0)$ as the predictor. As it is observed that motion vectors are highly correlated with the motion vectors of temporally and spatially adjacent blocks, the motion vectors of the collocated block in the previous frame or the adjacent blocks in the current frame are also considered as the predictor. In addition, the differentially increased/decreased motion vector named as *accelerator motion vector* is also used in [24].

In [21] BL MVP-based method is compared with four other predictors in terms of the entropy of bits representing the difference between predictors and the motion vectors generated using the full search and resultant PSNR. The BL MVP is shown to outperform other predictors in terms of the video quality and the entropy.

4.2.3 Concept of Activity

It was reported that FS BMA generally obtains less correlated motion vectors for the macro-blocks which are at the boundary of moving objects in a video sequence[22][23]. As depicted in figure 4-4 blocks C0, C1, C2 and C3 are blocks in enhancement layer corresponding to block B0 in the base layer. MV's of blocks C0-3 are less correlated to MV of B0 because motion of C0 and C2 differ from the motion of C1 and C3. Therefore it is necessary to extend the search range for blocks at boundary of moving objects.

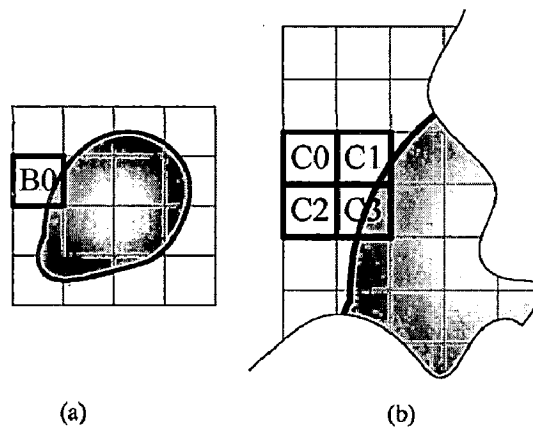


Figure 4-4 Grid on a sample object in (a) base layer and (b) enhancement layer; the rectangles with a bold line, B0 and C0-3, denote 4×4 blocks in the corresponding positions in the base layer and the enhancement layer, respectively.

Hence it is desired to identify those macro-blocks which are located in the object boundaries. Many gradient based operators have been employed to detect object boundary in object segmentation like Roberts, Prewitt and Rosenfeld methods. Because boundary operators are

based on the gradient magnitude, they often mistake a complicated texture in the scene for a moving object boundary or completely miss a moving object boundary when the gradient magnitude between the background and the boundary of the moving object is small. The moving object boundary prediction scheme based on activity excels others in terms of prediction accuracy [21] and computational complexity. The main purpose of moving object boundary detection in this activity based method is to judge whether a wider search range is necessary to achieve improved video quality than SR_{basis} or not before motion search, rather than to exactly extract the moving objects.

Activity A^l in layer l is defined as:

$$A^l = \max_i \left(\max(|mvd_x^l[i]|, |mvd_y^l[i]|) \right) \quad 0 \leq i \leq N-1 \quad (4.1)$$

where l denotes the layer index, $mvd_x^l[i]$ and $mvd_y^l[i]$ denotes x and y -component of the i th MVD of the corresponding MB in layer l , and N denotes the number of the MVDs. Because MVD shows how much the motion of current MB deviates from the MVP, which is either the BL MVP or the median MVP, MVD is used to predict the boundary of moving objects.

Regardless of the source of MVP, low activity usually means that the final MV is close to the MVP; this case is defined as “*regular motion*.” In other words, small search range is enough to search for the best-matched block if the block has a regular motion.

High activity occurs due to less correlated MVs at the boundary of moving objects. As a result, each block can be partitioned into two groups, a low-activity group and a high activity group. Activity regions are defined as follows:

- 1) Interior Region (IR) where MVs of the corresponding blocks in neighbouring layers are strongly correlated ($L_{MBB} \leq SR_{basis}$)
- 2) Boundary Region (BR) where the corresponding blocks in neighbouring layers are located near the boundary of moving objects, and MVs are weakly correlated ($L_{MBB} > SR_{basis}$).

4.2.4 Inter Layer Activity Model

Inter-layer activity model (ILAM) developed in [21] exploits the correlation of the mean activities between two neighbouring layers. It predicts the activity of the enhancement layer from that of the base layer with a linear equation

$$\hat{A}^l = \alpha * A^{l-1} + \beta \quad (4.2)$$

where \hat{A}^l is the predicted activity of the given MB in the enhancement layer (layer l), an inter-layer activity prediction factor, α , is the slope of ILAM denoted by a dashed line in Figure 4-5, an inter-layer activity prediction offset, β , is an intercept of ILAM, and A^{l-1} denotes the activity of the corresponding MB in the base layer (layer $l-1$).

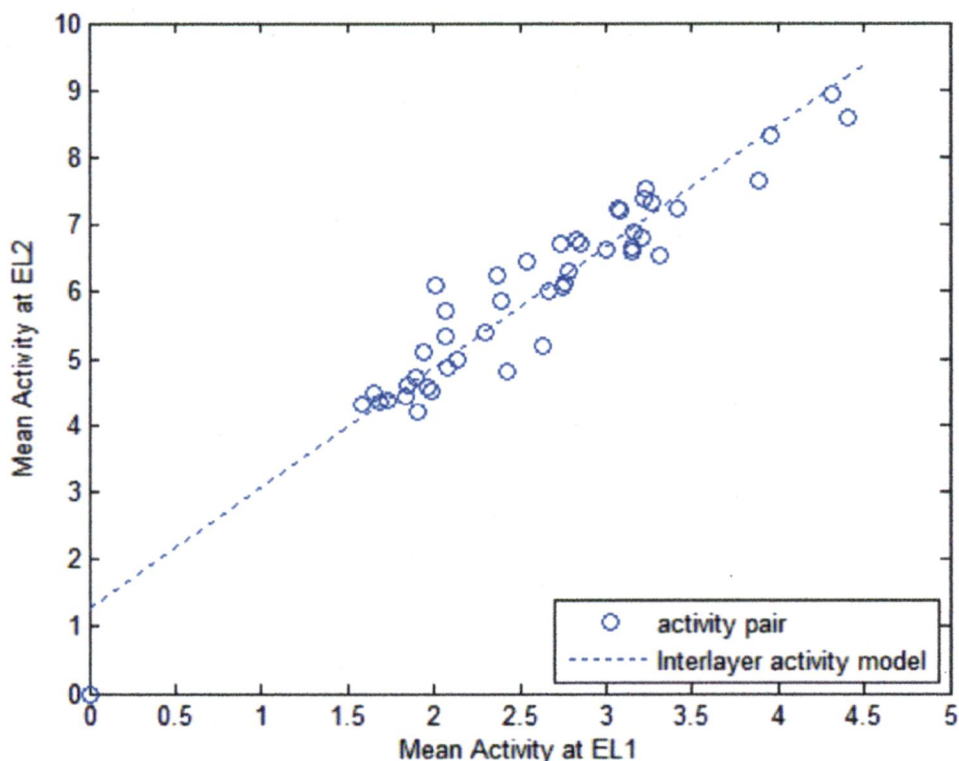


Figure 4-5 Activity plane representing pairs of the mean of activities between two neighbouring layers [base layer (BL) = CIF, enhancement layer (EL) = 4CIF] for video sequence ICE. The dashed line denotes inter-layer activity model with the given slope, $\alpha (= \Delta \bar{A}^l / \Delta \bar{A}^{l-1})$ and the given intercept, β where \bar{A}^{l-1} and \bar{A}^l denote the mean of activities over all MBs in a frame at the base and enhancement layer, respectively.

\bar{A}^l in Figure 4-5 denotes the mean of activities over all MBs in a frame at layer l . Values of α and β in equation 4.2 are obtained through the measurement with generic video sequences, such as *CREW*, *HARBOUR*, *ICE*, and *SOCCER* (45 frames with a SVC structure comprising three layers). Table 4-1 shows α , β , and *root mean square error (rmse)* measured with generic sequences. The second column shows the inter-layer activity prediction factor, α , for

the given sequence; the third column shows the inter-layer activity prediction offset, β ; the fourth column shows $rmse$, for given α and β .

Sequence	α	β	rmse
<i>SOCCER</i>	1.401	0.6523	0.8812
<i>CREW</i>	1.624	0.4845	0.6341
<i>ICE</i>	1.276	0.4706	0.4286
<i>HARBOUR</i>	1.086	0.5042	0.7192

Table 4-1 Inter-Layer Activity Prediction Factor, α , and Inter-Layer Activity Prediction Offset, β , for the given sequence, and the root mean square error, for given α and β , measured with generic sequences (45 frames on a SVC structure comprising three layers)

The value of α represents the coefficient of the assumed linear relationship between the activities in the neighbouring layers. Equation (4.2) is used before the motion search in the current layer to estimate the minimal search range to find the best motion vector without too much quality loss compared to the full search. Because the estimated search range is given as the product of α and the activity of the base layer, α affects both the computation time in ME and the video quality. α varies according to the given sequence while β is relatively steady. Therefore, α needs to be adjusted to satisfy the variation of the motion nature and the activity relationship between the neighbouring layers.

4.2.5 Procedure for simulation of Activity Based Motion Estimation (ABME) scheme

A. Overall Procedure of the Proposed Scheme

The activity-based ME (ABME) scheme takes one of the two paths, i.e., “ME for IR” and “ME for BR,” according to the activity of the base layer, A^{l-1} . At the beginning, the search range is given by inter-layer activity model (ILAM) using (4.2). If A^{l-1} is smaller than θ_{act} , the activity threshold, ABME takes ME for IR. Otherwise, AMBE takes ME for BR.

The final MV is chosen among the search results in terms of the PSNR cost. During the motion search, parameter α , the inter-layer activity prediction factor in (4.2) is adjusted where as θ_{act} is kept fixed at 8.

B. Search Centre

The search centre set is formed according to the given activity region. For IR base layer motion vector predictor (BL MVP) is used as search centre whereas for BR two motion vector predictors are used as search centre namely, BL MVP and median MVP.

The median MVP, is defined as

$$\hat{MV}_{med}^l = (MV_{left}^l, MV_{upper}^l, MV_{upper-right}^l) \quad (4.3)$$

where MV_{left}^l , MV_{upper}^l , and $MV_{upper-right}^l$ denote the MV of the left, upper, and upper-right block in the enhancement layer (layer l), respectively. BL MVP is obtained by up-scaling the MV of the base layer as mentioned in the previous section.

C. Adjustment of Inter-Layer Activity Prediction Factor, α

It is observed that α depends on the nature of motion in the scene and therefore, needs to be adjusted to the given sequence. The search range is not fixed but adjusted by (4.2) with a given α . After the motion search, it is checked whether the search range thus obtained is sufficient or not as follows. If the best point corresponding to the minimum SAD cost is close enough to the boundary of the search range, it is suspected that there exists some point with lower SAD cost than that point beyond the search range. On the other hand, if the best point is close enough to the predictor, the prediction is assumed to be quite accurate obviating the need for further checking of points far from the predictor.

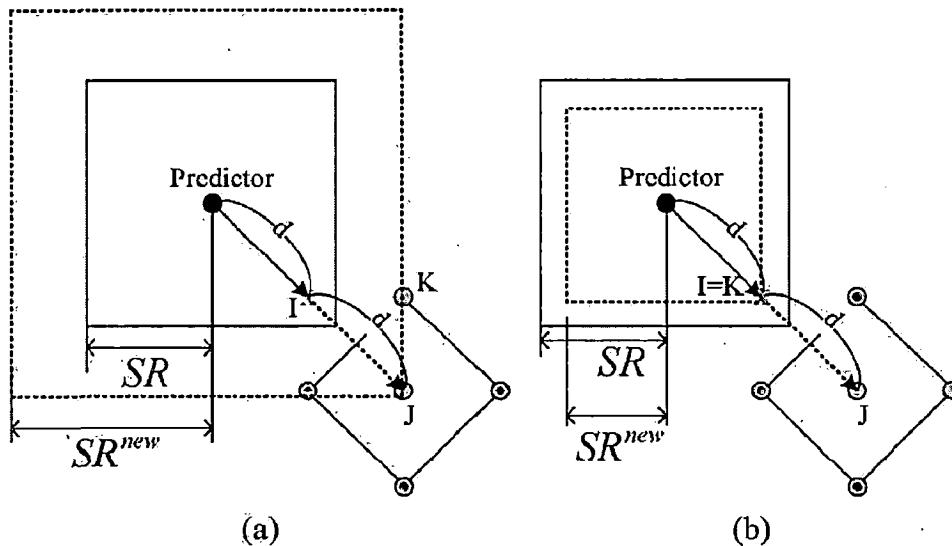


Figure 4-6 Optional check of diamond-shaped points (OCDSP), where SR is derived from (4.2). Point "I" denotes the point with the minimum SAD cost within the given search range, d denotes the distance between "Predictor" and point I, and point "J" denotes the centre point of the diamond-shaped search pattern whose distance from "Predictor" is twice as long as d . Five gray coloured circles denote optional check points in the diamond-shaped

search pattern, and point “K” denotes the point with the minimum SAD cost among six candidates, i.e., five optional check points and point I. SR^{new} denotes the required search range to cover point K (a) when point K is different from point I, and (b) when point K is identical to point I.

In Figure 4-6, a procedure called “optional check of diamond-shaped points (OCDSP)” as introduced in [21] is shown. The best point obtained by the motion search defined as point I and the centre point of the diamond pattern as point J, which is located twice as far as point I from the point denoted as “Predictor” along the direction of “Predictor-point I” vector. The radius of the diamond pattern is given as L_{MBB} , the length of the longer edge of the minimum bounding box (MBB) which covers both “Predictor” and point I. A new inter-layer activity prediction factor α' is obtained after the following steps defined as OCDSP.

1. Set the SAD cost of point I to $SADcost_I$.
2. Define the best point among five optional check points in the diamond pattern as point K.
3. Set the SAD cost of point K to $SADcost_K$.
4. If $SADcost_I < SADcost_K$ then point I is renamed as point K as shown in Figure 4-6 (b).
5. Get SR^{new} which minimally covers point K from “Predictor.”
6. Calculate α' deductively using (4.2) by using \hat{A}^l as the updated search range (SR^{new}) in (4.4)

$$\alpha' = \frac{SR^{new} - \beta}{A^{l-1}} \quad (4.4)$$

Here α is defined in two levels: in MB level (α_{MB}) and in frame level (α_{frame}). α_{MB} is computed by OCDSP after the completion of the motion search using the search range given by (4.2) with the previous value of α_{frame} , to be defined as α_{MB} . The mean of α' over all MBs in a frame is used to update α_{frame} . Initially α_{frame} is set to the maximum among values in Table 4-1 to support generic sequences. Activity Threshold, θ_{act} is kept fixed and its value is kept equal to SR_{basis} which is found to be 8 in previous section. If activity in layer l , A^{l-1} is lesser than θ_{act} then the corresponding MB belongs to IR otherwise to BR. For a MB in IR, search range is equal to SR_{basis} whereas for a MB in BR search range is obtained by (4.2).

4.2.6 Simulation Results for Activity Based Motion Estimation scheme

Here during the simulation to analyze only the effect of search algorithm, unlike motion estimation in video coding applications, backward motion estimation is performed between successive original frames for each of the three layers. This approach avoids the influence of

rate-distortion optimization and error propagation. Motion Estimation is performed by using only luminance component, and Sum of Absolute Difference (SAD) as the block distortion measure.

BL is taken as QCIF, first enhancement layer (EL1) as CIF and second enhancement layer (EL2) as 4CIF. Search centre for BL is zero MV i.e. search starts at the position of current MB in the reference frame. For EL1 search centre is determined by MVP that is, up-scaled version of BL motion vector. Before start of motion search for EL2 the activity at EL1 is available as computed from (4.1). Search centre for E2 is obtained based on the activity of corresponding MB in EL1. The search range for BL and EL1 is fixed to 8 and 16 respectively whereas search range for each MB in EL2 is determined based on its estimated activity \hat{A}^l as obtained from (4.2). Exhaustive search is performed within this search range. Fixed block size of 16×16 is used for each layer. PSNR is calculated between original and its corresponding motion compensated frame.

The simulation of this method is performed on generic video sequences to obtain the mean PSNR achieved at each layer, total number of SAD computations per MB for EL2 and total CPU time consumed in motion estimation process for all the three layers. PSNR, number of SAD computations per MB and CPU time are reported as the mean of their values for first 45 frames of given video sequences.

Video Sequence	PSNR BL (dB)	PSNR EL1 (dB)	PSNR EL2 (dB)	CBDM/MB EL2	T(BL+EL1+EL2) (sec)
<i>SOCCER</i>	36.5563	37.3857	37.7756	410.1988	272.4748
<i>CREW</i>	36.6851	39.7551	42.1084	632.5141	392.9842
<i>ICE</i>	39.7604	45.4240	48.3526	514.7710	313.9343
<i>HARBOUR</i>	31.9719	33.3836	35.3511	450.1835	291.0923

Table 4-2 Results as obtained through simulation of Activity Based Motion Estimation (ABME) method. PSNR for BL, EL1 and EL2 along with number of Computations of Block Distortion Measure per macro-block CBDM/MB for EL2 and total CPU Time (T) consumed during simulation for evaluation of motion vectors for all three layers.

Chapter 5 Proposed Motion Estimation Scheme for Scalable Video

Coding

Activity Based Motion Estimation scheme (ABME) as proposed by Sangkwon Na et al [21] as discussed in Chapter 4 uses motion vector predictor (MVP) and original motion vector (MV) for determining the activity of a particular MB in first enhancement layer (EL1). This activity is further used to estimate activity of corresponding MBs in second enhancement layer (EL2) to determine appropriate search range. The estimate of activity in EL2 is obtained using the Inter-Layer Prediction Model (ILPM) as specified by equation (4.2) where inter-layer prediction factor, α is adjusted dynamically for each frame based on the new search range obtained by Optional Check of Diamond Shaped Points (OCDSP) procedure as introduced in [21]. The search range thus obtained takes into account the activity of that MB in base layer and also the correlation of activity between base layer and enhancement layer. This search range is sufficient enough to encompass best matching MB in reference frame given the search centre.

Motion estimation in itself is a computationally expensive process and more so in case ME for scalable video coding as MV search is to be performed for each spatial resolution. The method proposed in [21] decreases the computational load on the encoder by specifying an appropriate search range for MBs in enhancement layer based on the information obtained during ME of lower resolution layers.

Any other ME scheme which is to be proposed must provide the same or better PSNR performance than [21] and should also be less computationally expensive. Keeping these constraints in mind we propose a modification to ABME scheme reported in [21].

5.1 Proposed Motion Estimation Scheme

In our proposed scheme we intend to obtain activity for EL1 more accurately by using half-pixel accurate ME for BL and EL1. Therefore, MVP and original MV of EL1 which are used for obtaining activity at EL1 are more accurately specified. The interpolation-free ME technique as proposed in [15] and discussed in chapter 3 is used to minimize the computational overhead introduced by half-pixel accurate ME.

The same inter-layer prediction model (ILPM) as proposed in [21] given by equation (4.2) is utilized to obtain an estimate of activity for a MB in EL2. The search range is obtained depending upon a given MB falls in interior region or boundary region and its estimated

activity. The search range thus obtained is more precise and is sufficient to contain the best matching MB within the search range.

The algorithm of the proposed scheme is as follows:

1. BL is taken as QCIF, first enhancement layer (EL1) as CIF and second enhancement layer (EL2) as 4CIF.
2. Search centre for BL is zero MV i.e. search starts at the position of current MB in the reference frame. The search range for BL is fixed to 8. Half-pixel MV refinement is performed on MVs of BL using interpolation-free motion estimation technique.
3. For EL1 search centre is determined by MVP that is, up-scaled version of half-pixel accurate BL motion vector. The search range for EL1 is fixed to 16. Also, half-pixel MV refinement is performed using interpolation free sub-pixel accurate ME technique for MV of EL1.
4. The activity at EL1, A^{l-1} is computed from equation (4.1) using its half-pixel accurate MV and its MVP obtained by up-scaling half-pixel accurate MV of BL.
5. Search centre for EL2 is obtained by up-scaling half-pixel accurate MV of EL1. The search range for each MB in EL2 is determined based on its estimated activity \hat{A}^l obtained from (4.2). The dynamic adjustment of inter-layer prediction factor, α is performed as already discussed in section 4.2.5.

Exhaustive search is performed within this search range. Fixed block size of 16×16 is used for each layer. PSNR is calculated between original and its corresponding motion compensated frame.

5.2 Simulation Results and Observations

Here the simulation is performed for the proposed scheme to analyze only the effect of search algorithm, unlike motion estimation in video coding applications, backward motion estimation is performed between successive original frames for each of the three layers. This approach avoids the influence of rate-distortion optimization and error propagation. Motion Estimation is performed by using only luminance component, and Sum of Absolute Difference (SAD) as the block distortion measure.

The simulation of proposed technique is performed to determine PSNR performance at each of the three layers, number of computations of block distortion measure per MB and total CPU time consumed in ME of all three layers. This CPU time includes the incremental CPU time consumed due to computational overhead caused by half-pixel MV refinement of BL and EL1. The three spatial resolutions used are QCIF: 176×144 , CIF: 352×288 and 4CIF: 704×576 for BL, EL1 and EL2 respectively.

Figure 5-1, 5-2 and 5-3 shows comparison of PSNR obtained by simulation of proposed ME scheme with that obtained by the simulation of ABME scheme [21] discussed in chapter 4 for BL, EL1 and EL2 respectively.

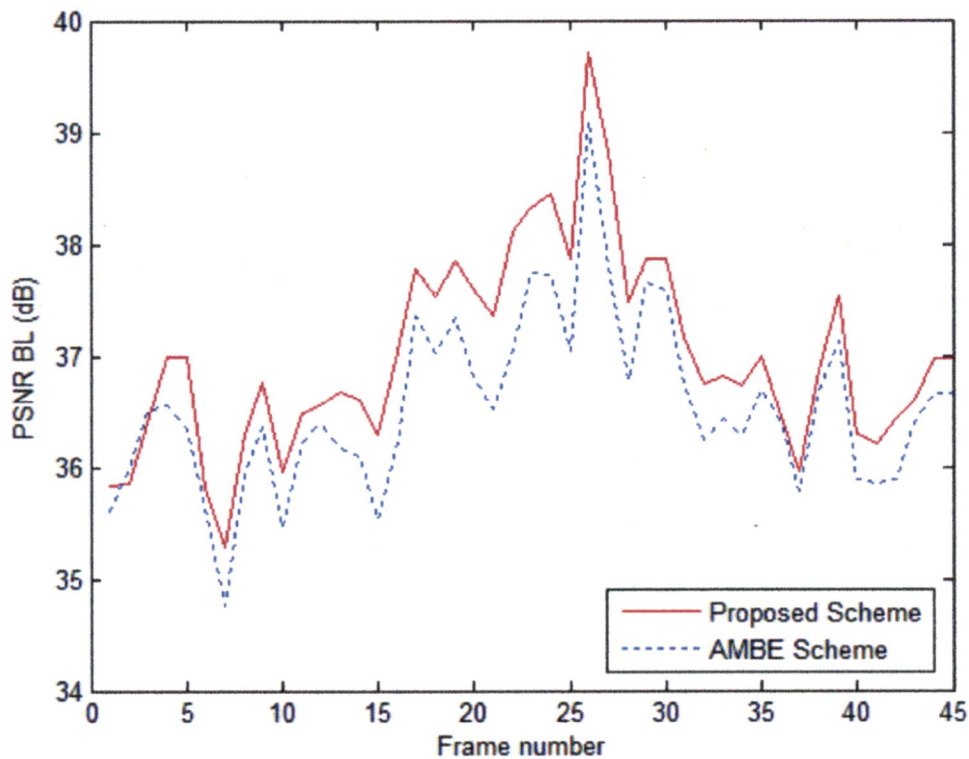


Figure 5-1 Comparison of PSNR obtained for BL by simulation of proposed ME scheme with that obtained by the simulation of AMBE scheme, Video Sequence: Soccer, 15 fps.

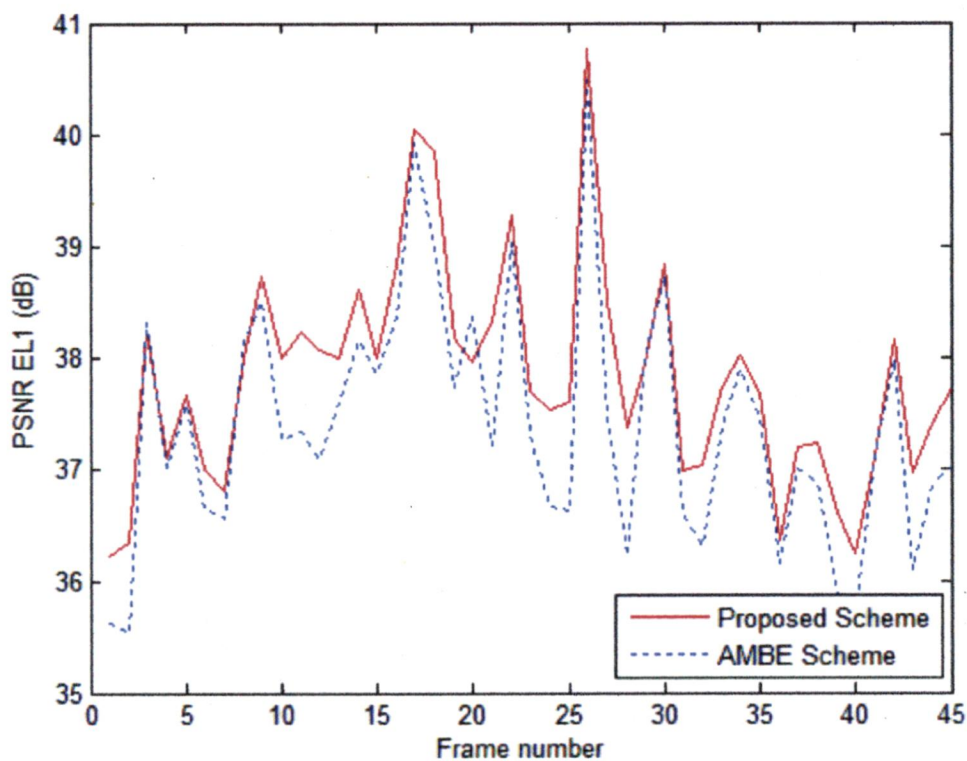


Figure 5-2 Comparison of PSNR obtained for EL1 by simulation of proposed ME scheme with that obtained by the simulation of AMBE scheme, Video Sequence: Soccer, 15 fps.

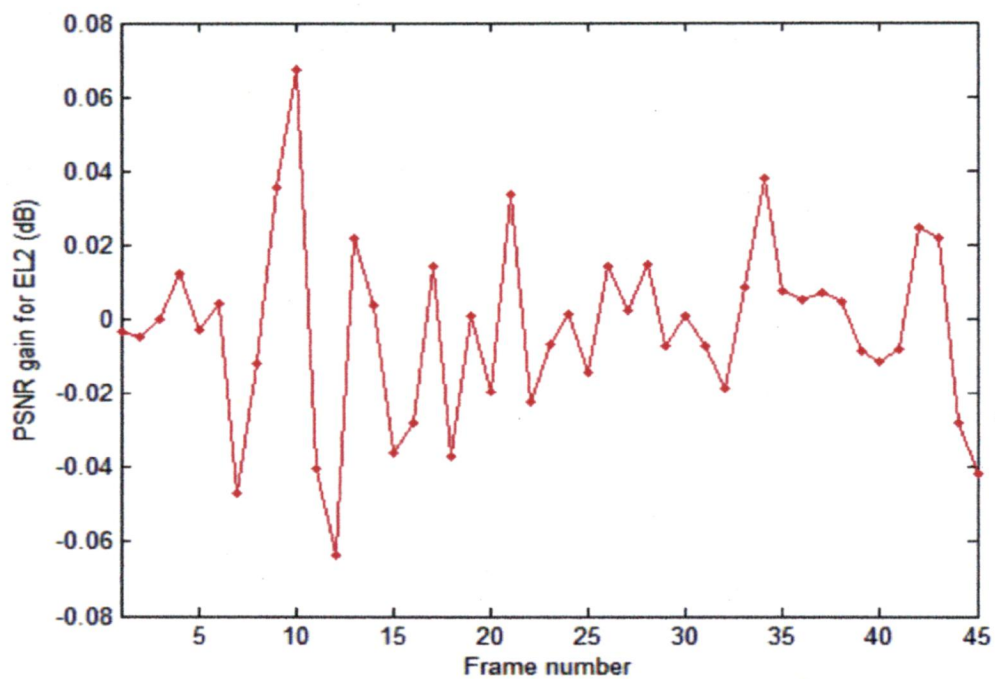


Figure 5-3 PSNR Gain obtained for EL2 by simulation of proposed ME scheme over the AMBE scheme, Video Sequence: Soccer, 15 fps.

From Figure 5-1 and 5-2 we observe that the PSNR is significantly improved for BL and EL1 because of availability of half-pixel accurate MVs at these layers in proposed scheme. From Figure 5-3 it is observed that PSNR performance of the proposed remains approximately the same in the proposed scheme.

Figure 5-4 shows the comparison of number of computations of Block Distortion Measure (BDM) per MB for EL2 obtained using the proposed scheme with that of ABME scheme.

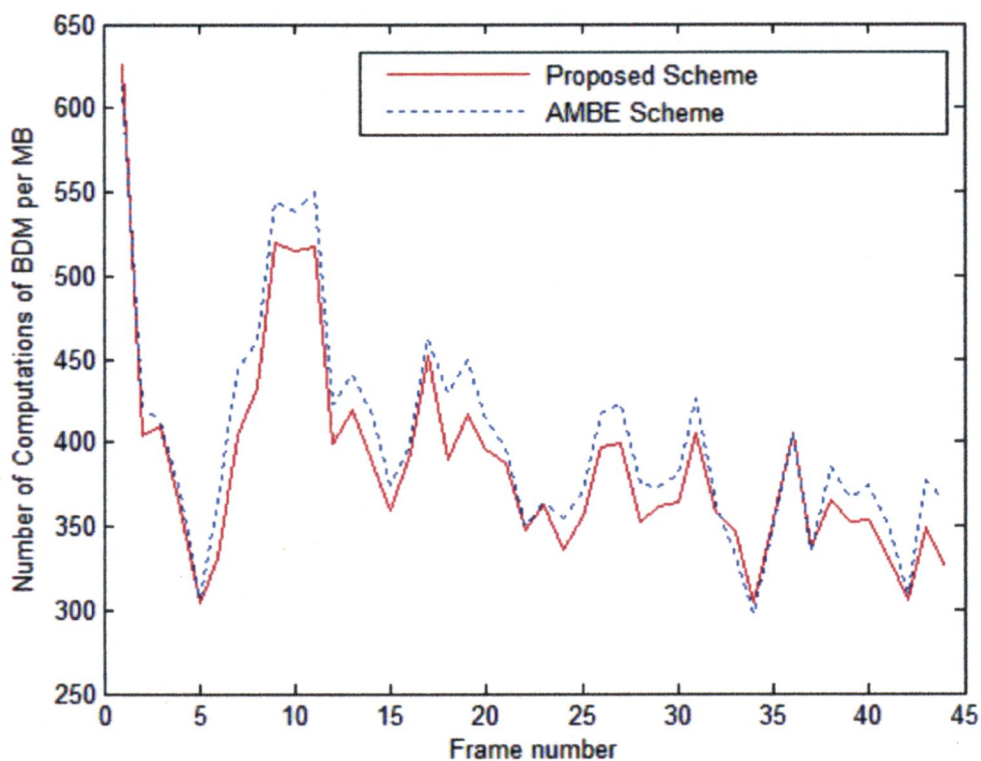


Figure 5-4 Comparison of number of computations of Block Distortion Measure (BDM) per MB for EL2 obtained using the proposed scheme with that of ABME scheme, Video Sequence: Soccer, 15 fps.

From Figure 5-4 we observe that there is decrease in number of computations of BDM per MB in EL2 by using the proposed technique whereas as seen from Figure 5-3 the PSNR performance in EL2 remains approximately the same.

Figure 5-5 shows the comparison of total CPU time consumed in ME for all three layers (including that spent in MV refinement of BL and EL1) obtained using the proposed scheme with that obtained using ABME scheme reported in [21].

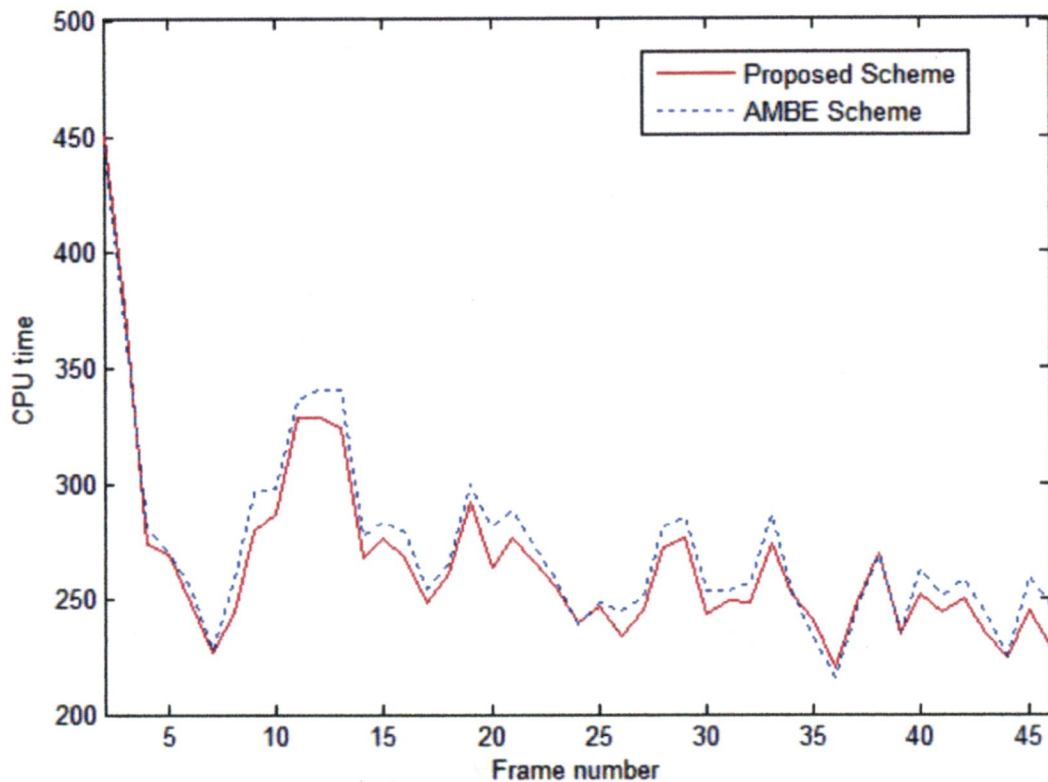


Figure 5-5 Comparison of total CPU time for Proposed scheme with that of ABME scheme, Video Sequence: Soccer, 15 fps.

From Figure 5-5 we observe that the total CPU time consumed in ME for proposed scheme is lesser than that of ABME scheme. This is because the computational overhead caused by the half-pixel accuracy motion vector refinement at BL and EL1 is offset by the reduction in computation at EL2 owing to more precise estimation of search range.

In Table 5-1 the mean of PSNR for all three layers, number of computations of BDM per MB (CBDM/MB) for EL2 and CPU time (T) consumed in ME of all three layers for first 45 frames of four generic video sequences is reported.

<i>Video Sequence</i>	<i>Quantity</i>	<i>Proposed ME scheme</i>	<i>ABME scheme</i>	<i>Difference</i>
<i>SOCER</i>	<i>PSNR R BL (dB)</i>	37.0036	36.5563	0.4473
	<i>PSNR EL1(dB)</i>	37.8197	37.3857	0.4340
	<i>PSNR EL2(dB)</i>	37.7729	37.7756	-0.0027
	<i>CBDM/MB for EL2</i>	394.9173	410.1988	-15.2815
	<i>CPU time,(sec)</i>	265.9106	272.4748	-6.5642
<i>CREW</i>	<i>PSNR BL (dB)</i>	36.8262	36.6851	0.1411
	<i>PSNR EL1(dB)</i>	40.0203	39.7551	0.2652
	<i>PSNR EL2(dB)</i>	42.1302	42.1084	0.0218
	<i>CBDM/MB for EL2</i>	618.3170	632.5141	-14.1971
	<i>CPU time,(sec)</i>	384.7109	392.9842	-8.2733
<i>ICE</i>	<i>PSNR BL (dB)</i>	40.1074	39.7604	0.3470
	<i>PSNR EL1(dB)</i>	46.0530	45.4240	0.6290
	<i>PSNR EL2(dB)</i>	48.3969	48.3526	0.0443
	<i>CBDM/MB for EL2</i>	502.8685	514.7710	-11.9025
	<i>CPU time,(sec)</i>	305.8060	313.9343	-8.1283
<i>HARBOUR</i>	<i>PSNR BL (dB)</i>	32.1200	31.9719	0.1481
	<i>PSNR EL1(dB)</i>	33.5239	33.3836	0.1403
	<i>PSNR EL2(dB)</i>	35.3594	35.3511	0.0083
	<i>CBDM/MB for EL2</i>	445.3571	450.1835	-4.8264
	<i>CPU time,(sec)</i>	284.5788	291.0923	-6.5135

Table 5-1 Comparison of Proposed ME scheme with ABME scheme based on PSNR at all layers, number of computations of BDM per MB (CBDM/MB) and CPU time (T)

From Table 5-1 we can see that there is a PSNR improvement achieved at BL and EL1 for all four sequences using the proposed scheme. Also PSNR at EL2 remains almost unchanged. Whereas there is a reduction in overall CPU time consumed for ME of all three layers. This is because the computational overhead caused by half-pixel accuracy MV refinement at BL and EL1 is offset by the computational reduction at EL2 caused by the availability of a more appropriate search range.

Chapter 6 Conclusion

Motion estimation used to remove temporal redundancy in video encoders, is a computationally expensive process. Many techniques have been proposed till date to reduce computational complexity of motion estimation while providing good estimation performance.

In this thesis, we have compared fast block matching motion estimation algorithms in terms of PSNR and computational complexity. Also, a fast interpolation-free ME technique is implemented and its performance is compared with conventional approach. Activity based ME scheme is also implemented and its simulation results for PSNR performance and computational complexity are evaluated.

Also, we have proposed a new motion estimation scheme for scalable video coding. This technique is based on estimating search range accurately for motion estimation at enhancement layer using the information obtained during motion estimation at lower layers.

The proposed scheme decreases the overall CPU time required for motion estimation whereas PSNR at BL and EL1 is improved significantly and PSNR at EL2 remains approximately the same as compared to ABME scheme proposed in [21].

Future work includes the implementation of proposed scheme for standard encoder like JSVM and measure the rate-distortion performance of the proposed scheme.

Works Cited

- [1] "Video codec for audiovisual services at $p \times 64$ kbit/s," International Telecommunication Union- Telecommunications. (ITU-T), Geneva, Switzerland, Recommendation H.261, 1993.
- [2] "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, Part 2: Video," International Standards Org./ International Electrotechnical Commission (ISO/IEC), ISO/IEC 11 172-2.
- [3] "Generic coding of moving pictures and associated audio information— Part 2: Video," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 13 818-2 (identical to ITU-T Recommendation H.262).
- [4] "Coding of audiovisual objects—Part 2: Visual," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), 14 496-2.
- [5] "Coding of audiovisual objects—Part 10: Advanced video coding," Int. Standards Org./Int. Electrotech. Comm. (ISO/IEC), ISO/IEC 14 496-10 (identical to ITU-T Recommendation H.264).
- [6] Ohm, J.-R.; , "Advances in Scalable Video Coding," *Proceedings of the IEEE* , vol.93, no.1, pp.42-56, Jan. 2005.
- [7] I. E. G Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*, Wiley publications, West Sussex, England 2003.
- [8] S. Aramvith and M. T. Sun, "MPEG-1 and MPEG-2 Video Standards," *Handbook of Image and Video Processing*, Second Edition, Elsevier Academic Press, 2005.
- [9] J Chen, U. V. Koc and K. G. R. Liu, *Design of Digital Video Coding Systems: A Complete Compressed Domain Approach*, Marcel Dekker, Inc., New York, 2002.
- [10] B.Liu and A.Zaccarin, "New fast algorithms for the estimation of block motion vectors," *IEEE Transactions on Circuits and Systems for Video technology*, Vol.3, pp.440–445,Dec 1995.
- [11] Xuan Jing and Lap-Pui Chau , "An efficient three-step search algorithm for block motion estimation," *IEEE Transactions on Multimedia*, vol.6, no.3, pp. 435- 438, June 2004.

- [12] Lai-Man Po and Wing-Chung Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.6, no.3, pp.313-317, Jun 1996.
- [13] Shan Zhu and Kai-Kuang Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Transactions on Image Processing*, vol.9, no.2, pp.287-290, Feb 2000.
- [14] R. A. Manap, S. S. S Ranjit, A. A. Basari and B. H. Ahmad, "Performance Analysis of Hexagon-Diamond Search Algorithm for Motion Estimation," *2nd International Conference on Computer Engineering and Technology (ICCET), 2010*, vol.3, no., pp. 155-159, 16-18 April 2010.
- [15] P.R. Hill, T.K. Chiew, D.R. Bull, C.N. Canagarajah, "Interpolation Free Subpixel Accuracy Motion Estimation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.16, no.12, pp.1519-1526, Dec. 2006.
- [16] S. Dikbas, T. Arici and Y. Altunbasak, "Fast Motion Estimation With Interpolation-Free Sub-Sample Accuracy," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.20, no.7, pp.1047-1051, July 2010.
- [17] Liang-Wei Lee, Jing-Fa Wang, Jau-Yien Lee and J.-D. Shie, "Dynamic search-window adjustment and interlaced search for block-matching algorithm," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.3, no.1, pp.85-87, Feb 1993.
- [18] T. Yamada, M. Ikekawa and I. Kuroda, "Fast and accurate motion estimation algorithm by adaptive search range and shape selection," *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP '05)*, pp. 897- 900, Vol. 2, 18-23 March 2005.
- [19] Tian Song, K. Ogata, K. Saito and T. Shimamoto, "Adaptive Search Range Motion Estimation Algorithm for H.264/AVC," *IEEE International Symposium on Circuits and Systems, 2007. ISCAS 2007*, pp.3956-3959, 27-30 May 2007.
- [20] Zhenxing Chen, Qin Liu, T. Ikenaga and S. Goto, "A motion vector difference based self-incremental adaptive search range algorithm for variable block size motion estimation," *15th IEEE International Conference on Image Processing, 2008. ICIP 2008*, pp.1988-1991, 12-15 Oct. 2008.

- [21] Sangkwon Na and Chong-Min Kyung , "Activity-Based Motion Estimation Scheme for H.264 Scalable Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.20, no.11, pp.1475-1485, Nov. 2010.
- [22] Bo Shen, I.K. Sethi and B. Vasudev, "Adaptive motion-vector resampling for compressed video downscaling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol.9, no.6, pp. 929-936, Sep 1999.
- [23] M. J. Chen, M.-C. Chu and S. Y. Lo, "Motion vector composition algorithm for spatial scalability in compressed video," *IEEE Transactions on Consumer Electronics*, vol. 47, no. 3, pp. 319–325, Aug. 2001.
- [24] H. Y. C. Tourapis, and A. M. Tourapis, "Fast motion estimation within the H.264 codec," *Proceedings of International Conference on Multimedia and Expo, 2003. ICME '03.. 2003* , vol.3, no., pp. 517-520, 6-9 July 2003.