Surveillance Centric Coding

Muhammad Akram

Department of Electronic Engineering
Queen Mary, University of London

Thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

2011

To my Parents and Family

Abstract

The research work presented in this thesis focuses on the development of techniques specific to surveillance videos for efficient video compression with higher processing speed. The Scalable Video Coding (SVC) techniques are explored to achieve higher compression efficiency. The framework of SVC is modified to support Surveillance Centric Coding (SCC). Motion estimation techniques specific to surveillance videos are proposed in order to speed up the compression process of the SCC.

The main contributions of the research work presented in this thesis are divided into two groups (i) Efficient Compression and (ii) Efficient Motion Estimation. The paradigm of Surveillance Centric Coding (SCC) is introduced, in which coding aims to achieve bit-rate optimisation and adaptation of surveillance videos for storing and transmission purposes. In the proposed approach the SCC encoder communicates with the Video Content Analysis (VCA) module that detects events of interest in video captured by the CCTV. Bit-rate optimisation and adaptation are achieved by exploiting the scalability properties of the employed codec. Time segments containing events relevant to surveillance application are encoded using high spatio-temporal resolution and quality while the irrelevant portions from the surveillance standpoint are encoded at low spatio-temporal resolution and / or quality. Thanks to the scalability of the resulting compressed bit-stream, additional bit-rate adaptation is possible; for instance for the transmission purposes. Experimental evaluation showed that significant reduction in bit-rate can be achieved by the proposed approach without loss of information relevant to surveillance applications.

In addition to more optimal compression strategy, novel approaches to performing efficient motion estimation specific to surveillance videos are proposed and implemented with experimental results. A real-time background subtractor is used to detect the presence of any motion activity in the sequence. Different approaches for selective motion estimation, GOP based, Frame based and Block based, are implemented. In the former, motion estimation is performed for the whole group of pictures (GOP) only when a moving object is detected for any frame of the GOP.

While for the Frame based approach; each frame is tested for the motion activity and consequently for selective motion estimation. The selective motion estimation approach is further explored at a lower level as Block based selective motion estimation. Experimental evaluation showed that significant reduction in computational complexity can be achieved by applying the proposed strategy. In addition to selective motion estimation, a tracker based motion estimation and fast full search using multiple reference frames has been proposed for the surveillance videos.

Extensive testing on different surveillance videos shows benefits of application of proposed approaches to achieve the goals of the SCC.

Acknowledgements

It is my greatest pleasure to express my sincere appreciation and gratitude to my supervisor Prof. Ebroul Izquierdo for his beneficial discussions, encouragement, valuable advice and guidance throughout the course of this work. I would also like to thank Dr. Naeem Ramzan and Dr. Toni Zgaljic for their positive and constructive comments during my research.

It was a pleasure to work with many brilliant people in the MMV Group during my stay at the lab: Dr. Ivan Damnjanovic, Dr. Giuseppe Passino, Dr. Karishna Chandramouli, Dr. Tomas Piatric, Dr. Tijana Janjusevic, Dr. Qianni Zhang, Mr. Eduardo Peixoto, Miss Virginia Fernandez Arguedas, and Mr. Stefano Asioli. Thanks to all of them for their support and patience during my PhD.

My special thanks go to Dr. Masood Ur Rehman, Dr. Nadeem Anjum, Mr. Qammer Hussain Abbassi, Mr. Lasantha Seneviratne, Mr. Sander Koelstra, Mr. Fahad Danyal and Dr. Murtaza Taj for their warm friendship and constant support.

This section will be incomplete without acknowledging full financial support of Government of Pakistan through University of Engineering & Technology, Lahore Pakistan.

Finally, I would also take this opportunity to thank my parents, brothers and sisters for their continuous love and support throughout my PhD and life.

Table of Contents

Chapter 1…..……..…….……1
1. Introduction…….…….……..1
1.1. Motivation……..…….…….……...5
1.2. Scope of the thesis…….…….……7
1.3. Contributions and Structure of the Report..…….…….……..8
1.3.1. Efficient Compression……8
1.3.2. Efficient Motion Estimation……...9
1.3.3. Thesis Structure……10
Chapter 2……...…...12
2. Background……..12
2.1. Rate-Distortion Theory……12
2.2. Rate-Distortion Optimisation for Video Coding…….……14
2.2.1. Distortion Measures……15
2.2.2. Lagrangion Optimisation……16
2.3. State-of-the-Art Work……...18
Chapter 3……..…20
3. Block-Based Video Coding……...…….……20
3.1. Intra-Prediction Mode..……..…….……..22
3.2. Inter-Prediction Mode. ……..……..…….24
3.3. Transform and Quantisation.....……....25
3.4. Entropy Coding……..……..….26
3.5. Scalable Video Coding…….27
3.6. Wavelet-Based Scalable Video Coding……28
3.7. Temporal Transform…….30
3.8. Spatial Transform…….32
3.9. Scalability support in the bit-stream…….33

Chapter 4……..36
4. Object-Based Video Coding…….……...……36
4.1. Object Shape Coding……38
4.1.1. Binary Shape Coding……...38
4.1.2. Grey scale Shape Coding…….39
4.2. Foreground Coding……...39
4.2.1. Motion Estimation and Compensation…….39
4.2.2. Boundary Macroblock Padding……40
4.2.3. Exterior Macroblock Padding……..40
4.2.4. Texture Coding…….41
4.3. Background Coding……..42
Chapter 5……..43
5. Towards Surveillance Centric Coding…….……...…….43
5.1. Architectural Modification……...44
5.1.1. GOP Selector……44
5.1.2. GOP Collector……..44
5.1.3. GOP Analysis……...45
5.2. Methodology…….46
5.2.1. Encoder…….46
5.2.2. Extractor……...47
5.2.3. Decoder……48
5.3. Model of the System…….48
5.3.1. Encoder…….48
5.3.2. Extractor……...49
5.3.3. Decoder……50
5.4. Functionality Evaluation…….……..…….…...50
5.5. Surveillance Centric Coding…….53
5.5.1. Event Based Coding of Surveillance Videos……...54
5.5.2. Background Subtraction for Motion tracking……..54
5.5.3. Proposed System……..56

5.6. Performance Results……..……57
5.7. Foreground Based SCC……63
5.8. Methodology……64
5.9. Experimental Results for FBSCC……66
5.10. Related Test Results…….……70
Chapter 6……..75
6. Selective Motion Estimation……75
6.1. Introduction……...75
6.2. Selective Motion Estimation…….76
6.2.1. Real-Time Background Subtraction……77
6.2.2. Selection Policy……77
6.3. Experimental Results……79
6.4. Selective Block Search…….84
6.4.1. Block Selection Policy…….84
6.5. Experimental Results……86
6.6. Tracker Based Motion Estimation ……...90
6.7. Experimental Results ……...……92
6.8. Conclusions……...…98
Chapter 7……100
7. Fast Full Search for Motion Estimation …..…….100
7.1. Successive Elimination Algorithm…….101
7.2. Fast Full Search in H.264……...102
7.3. Multiple Reference Frames Motion Estimation…….104
7.4. Fast Multiple Reference Frames Motion Estimation……..104
7.4.1. The Algorithm……105
7.5. Experimental Results……..110
7.6. Conclusions…….114
Chapter 8 ……..115
8. A Multi-Pattern Search Algorithm…….115
8.1. Introduction…….115

8.2. Multi-Pattern Search…. ……116
8.3. Simulation Results……..120
8.4. Conclusions……..……...123
Chapter 9……125
9. Conclusions…… ……...125
9.1. Conclusions……125
9.2. Key Contributions..……128
9.2.1. Efficient Compression……128
9.2.2. Efficient Motion Estimation……...129
9.3. List of Publications……131
References……..133

List of Abbreviations

SVC	Scalable Video Coding
SCC	Surveillance Centric Coding
VCA	Video Content Analysis/Analyser
CCTV	Closed Circuit TV
MPEG	Moving Pictures Expert Group
AVC	Advance Video Coding
RD	Rate Distortion
GOP	Group of Pictures
ME	Motion Estimation
MB	Macroblock
SEA	Successive Elimination Algorithm
MCTF	Motion Compensated Temporal Filtering
DWT	Discrete Wavelet Transform
HVS	Human Visual System
SSD	Sum of Squared Difference
SAD	Sum of Absolute Difference
PSNR	Peak Signal to Noise Ratio
MSE	Mean Square Error
JVT	Joint Video Team
VCEG	Video Coding Expert Group
ITU-T	International Telecom Union – Telecom standard
ISO/IEC	International Standard Organisation/International Electrotechnical Commission
NAL	Network Abstraction Layer
CABAC	Context Adaptive Binary Arithmetic Coding
CAVLC	Context Adaptive Variable Length Coding
JPEG	Joint Photographic Experts Group

CGS	Coarse Grain Scalability
VOP	Video Object Plane
GOV	Group of VOP
CAE	Context based Arithmetic Encoding
DCT	Discrete Cosine Transform
CIF	Common Intermediate Format
VM	Verification Model
JM	Joint Model
JSVM	Joint Scalable Video Model
BMA	Block Matching Algorithm
BDM	Block Distortion Measure
MRFME	Multiple Reference Frame Motion Estimation
MAE	Mean Absolute Error
CDHS	Cross Diamond Hexagonal Search

Chapter 1

Introduction

The research presented in this thesis focused on diverse coding techniques specific to surveillance videos. The foremost aim of this work is to propose techniques to improve the storage capacity and bandwidth utilisation with less computational complexity. The proposed coding techniques to improve compression and processing efficiency have been described to achieve better performance compared to the conventional techniques.

One of the major building blocks of the modern digital video surveillance architecture is digital video coding. Its main role is to decrease the quantity of information essential to represent the original image sequence. Video coding techniques offer a compressed bit-stream representing the identical perceptual information with much less data for any given input video with a particular image frame rate and resolution. Superior compression performance can be accomplished with sophisticated video coding methods and/or bringing in visual artefacts.

Once compression is executed, the consequent bit-stream can be resourcefully propelled through a digital network or stored on a device. When a client needs to exhibit it or explore its contents, a decoding procedure requires to be applied on the compressed bit-stream. The decoding course of action recreates the original input video at its original resolution and frame rate.

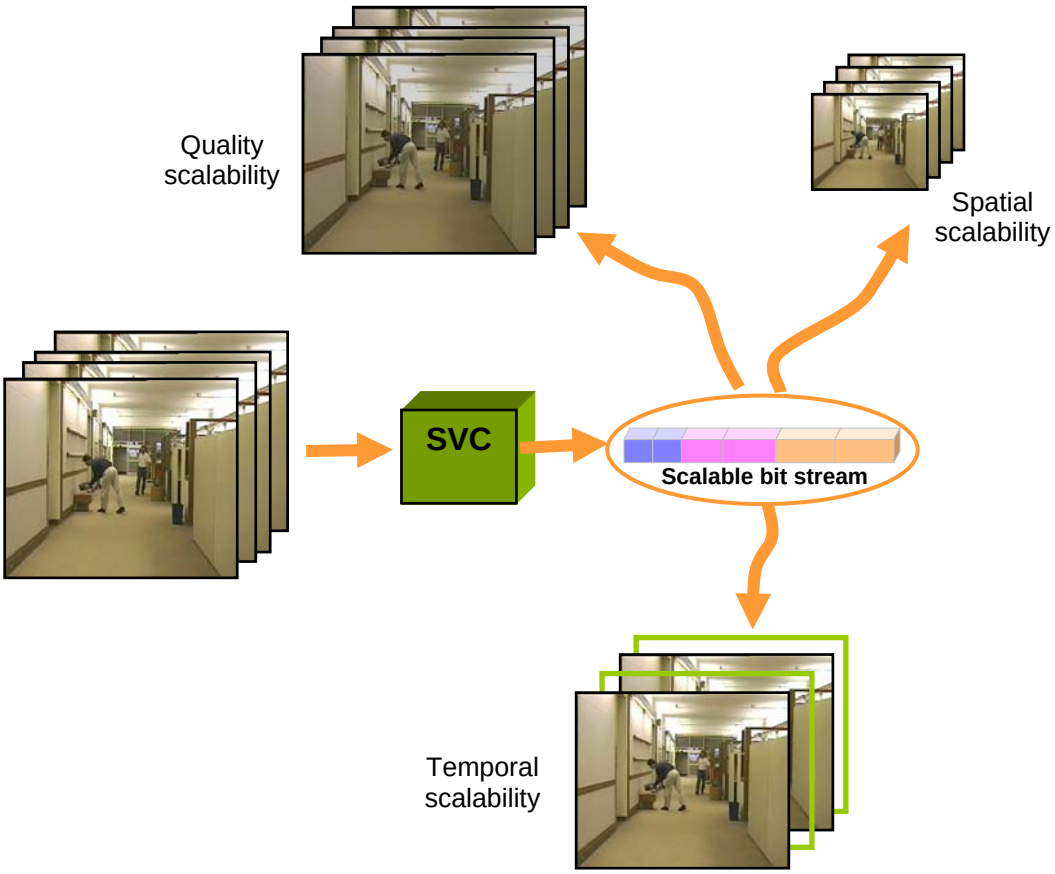


Figure 1: Scalable video coding

Scalable Video Coding (SVC) offers the equivalent compression functionality expressed above and is shown in Figure 1. Further more, the bit-stream is arranged with a hierarchical structure that facilitates a user to effortlessly pull out only a subpart of the data enclosed in the bit-stream while still being able to decode the original input video but at a lower frame rate and/or spatial resolution. The recursive application of this approach on a new bit-stream removed out of the original bit-stream can be utilised to carry out the process of successive extractions consequent to always lower resolutions.

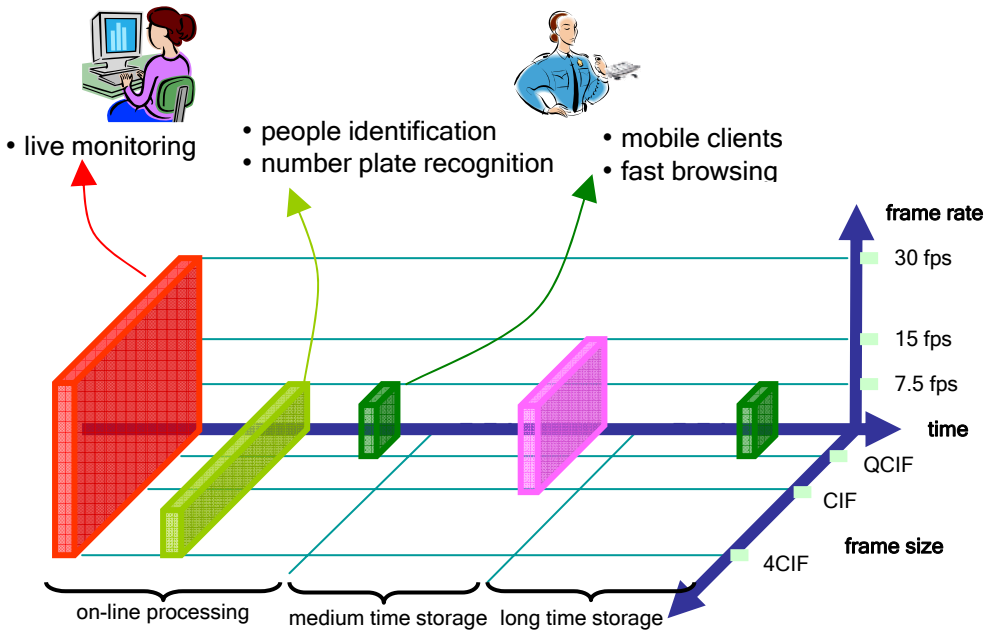


Figure 2: Surveillance application of SVC

There are numerous kinds of scalability and each of them can be available with different granularity. The most frequent type of scalability is the temporal scalability. For example, if the original image sequence consisted of 30 frames per second, temporal scalability would facilitate a user to decode a subpart of the bit-stream reconstructing a sequence with 15 frames per second or a lower number of frames per second. Spatial scalability is related to the opportunity to generate a bit-stream corresponding to the smaller spatial resolutions, for example, by decoding images with a resolution of CIF (common intermediate format: 352×288) out of an image sequence initially encoded at 4CIF (704×576). Quality scalability represents the possibility to decode the bit-stream at a lower quality. In this scenario, the temporal and spatial resolutions continue to be the same, but the recreated image sequence will emerge having additional artefacts or less details. In short, Scalable Video Coding (SVC) offers an exclusive representation of one image sequence permitting instantaneous access to the scene at different scales: spatial, temporal and quality. One of the many application scenarios of SVC is shown in Figure 2 where benefits of SVC for visual surveillance are evident.

Surveillance Centric Coding (SCC) is based on the SVC frame-work due to its potential benefits for surveillance applications. The main goal of this thesis is to develop such surveillance video specific coding techniques which offer a higher

compression ratio so that storage and transmission resources may be utilized more efficiently. In addition to a better compression performance, some techniques to perform a quicker compression process are also focused for the SCC. All the developed techniques to accomplish these tasks are presented in consequent chapters of this thesis.

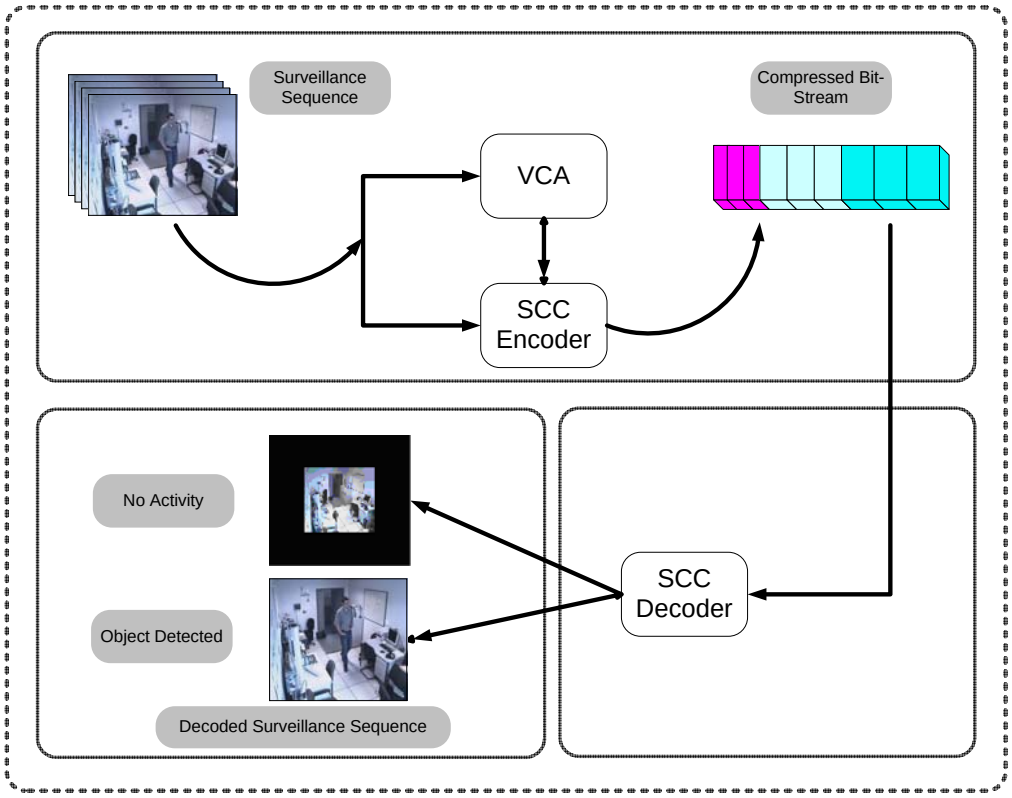


Figure 3: Surveillance centric coding system

An overview of the SCC system is shown in Figure 3. The Video Content Analyser (VCA) is used to detect any important activity corresponding to surveillance specific information. Based on this information, the reconstructed image resolution is specified. Thus, a lot of storage space is saved for the case when there is no motion activity. Now for the case of improving processing time, the focus of the attention goes to the motion estimation module due to its high processing power consumption. A high level system overview is illustrated in Figure 4. The process of motion estimation is driven on the basis of the information provided by the motion detection module. The final compressed bit-stream is available after the process of Entropy

Coding. The detailed discussion on the SCC system is presented in the dedicated chapters.

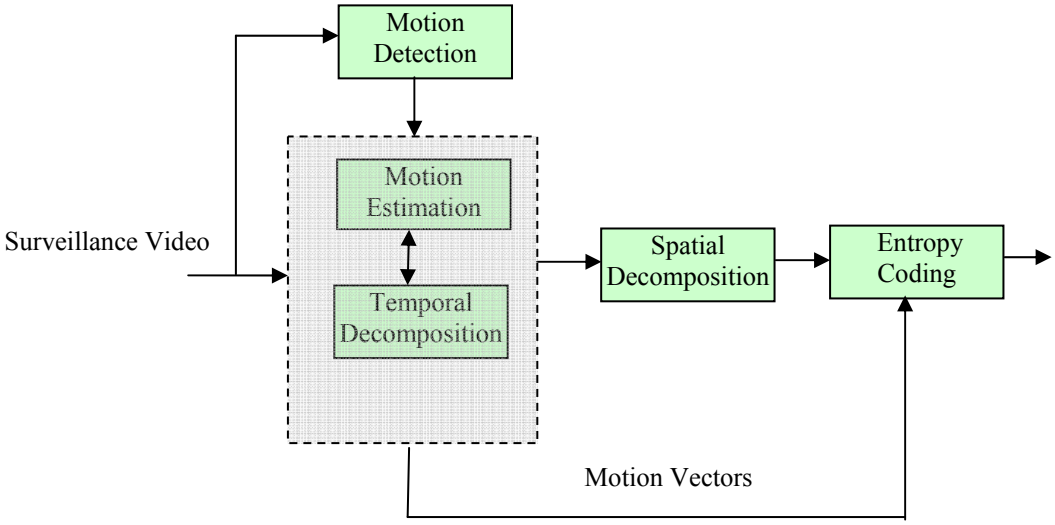


Figure 4: Surveillance centric coding system: Motion Estimation

1.1 Motivation

Security has been a critical issue in the world. Different security techniques have always been deployed according to different scenarios. Over the past decade or so, video-Surveillance has evolved as the most widely used security system today. It has seen great success and growth in the rate of deployment during recent times. Surveillance cameras can be found at almost every security-sensitive point as shown in Figure 5. The massive deployment of Closed Circuit Television (CCTV) cameras generates an enormous amount of data as can be observed in Figure 6. As the data keeps on increasing with 24/7 CCTV operation, the problem of data management escalates. The situation is further jeopardized when multi-view cameras are installed in highly security-sensitive locations.

Many researchers are drawing inspiration from this scenario and exploring new avenues in the CCTV system related research and areas like object recognition, object–based video segmentation, target detection and visual tracking etc. Apart from the problem of monitoring extracting video surveillance information, the huge

amount of data imposes a problem of storage and transmission with limited resources like storage space and channel bandwidth.

In surveillance applications video captured by CCTV is usually encoded using conventional compression technology, such as MPEG-1/2 or H.264/AVC. These systems encode the video signal regardless the significance of events in the video. As an example, in many surveillance situations the scene remains essentially static for seconds and even minutes in some cases. During these periods of time nothing interesting happens from the surveillance standpoint, and the video resembles a still picture for long periods of time with no other activity than random environmental motion. This is the case of surveillance in metro stations during night hours or private car parking where the usual events are cars coming and leaving from time to time. When such videos are compressed using conventional coding techniques, each frame receives an equal level of importance. The conventional video compression techniques do not make a distinction between a frame having no special information and a frame having an object of interest with some motion activity.

Figure 5: Extensive deployment of surveillance cameras

The primary goal of developing any codec is to achieve highest compression while maintaining best possible visual quality. Therefore, all the modules and processes of the codec are optimised to achieve this goal. Thus, the video codecs available today have been optimised for visual sensitive applications. These codecs are not very efficient for surveillance systems. Therefore, we need to develop a coding scheme which is specific to surveillance system to improve the coding efficiency. Consequently, the utilisation of storage space and transmission of the channel bandwidth, especially in wireless channel, will improve.

Figure 6: CCTV system and control room

In Surveillance Centric Coding (SCC), the framework of scalable video coding shall be utilised for Rate-Distortion (RD) optimisation specific to surveillance videos. The SCC will be flexible in changing its coding parameters according to the significance of the video events. The SCC shall also code different regions of each frame with different fidelity. Moreover, the SCC shall have lower complexity compared to the current standards.

1.2 Scope of the Thesis

The major goal of the work presented in this thesis is to develop techniques specific to surveillance videos to achieve better compression at reduced computational complexity. The obvious characteristics of surveillance videos include static background or static cameras and a high importance for the video segments containing any event of interest like car entering a parking area. Such characteristics motivate to explore the Scalable Video Coding (SVC) frame work and computer vision techniques relevant to surveillance videos like the VCA. In order to reach to a meaningful conclusion, following research questions need to be addressed.

Will the SVC be helpful to achieve byte savings? Does the SVC framework have support to achieve goals of Surveillance Centric Coding (SCC)? How to utilise the SVC framework if it does not have a support for the SCC? Are there any possibilities of integrating computer vision techniques developed for surveillance videos in the

framework of the SCC? How would computer vision techniques help to realise the SCC? What will be the effect of computer vision techniques on the complexity of the SCC? How will the negative points of computer vision techniques affect? What will be the effect of different scenarios on the deployment of the SCC?

This is worth-mentioning that the work presented in this thesis does not deal with the computer vision techniques and does not address the problems related to them. The computer vision techniques are explored and utilised to achieve the byte saving and reduction in computational complexity in the SCC. The thesis deals with the effects of integrating computer vision techniques in the framework of the SCC.

1.3 Contributions and Structure of the Thesis

According to the objectives laid down for the PhD research, some techniques have been developed for efficient compression and efficient motion estimation of surveillance videos. These techniques are discussed in the subsequent chapters of this thesis. The corresponding major contributions are discussed in the following subsections.

1.3.1 Efficient Compression

Scalable video coding (SVC) framework is selected as a basis to carry out the development of Surveillance Centric Coding (SCC) techniques after distinguishing the potential benefits of using the SVC techniques for surveillance videos. In order to deal at the Group of Pictures (GOP) level, the GOP dependency on the preceding and following GOPs is removed. The second step is to get the ability to deal with each GOP according to the SCC requirements while still maintaining the SVC properties. So, the communication linkage between Video Content Analysis (VCA) module and single GOP is established. The analysis of each GOP generated by the VCA under the requirements of SCC is utilised to exploit the SVC properties of each GOP. Thus, the bit-stream generated by the SCC consists of GOPs with different scalability features to compress the surveillance videos with higher compression efficiency while no compromise is made on the important information from the surveillance standpoint.

After the implementation of SCC paradigm, another novel approach is proposed to improve the compression efficiency. In this approach, foreground objects are detected

by the VCA by forming rectangular windows around objects. The first frame of the sequence is used as background and rest of the frames contain only the foreground pixels while the background pixels are set to zeros. This shows efficient compression; but due to use of block based coding approaches, lack of sharpness at the object edges is observed. The major advantage of using this approach is its implementation in the SCC framework. Thus, in addition to avoiding shape coding and other object based coding techniques, the scalabilities features are inherited through the SCC framework offering the potential of improving the compression efficiency through exploiting the scalability features in each GOP.

1.3.2 Efficient Motion Estimation

As the motion estimation (ME) is the most processing intensive part of a codec; efficient techniques to perform fast motion estimation are explored. A novel approach of performing selective motion estimation is proposed where object detection information generated by the VCA is used to flag the frames which do not have any moving object. Based on this analysis, different selective motion estimation approaches were proposed which included: (i) GOP level selective motion estimation, (ii) Frame level selective motion estimation and (iii) block level motion estimation.

After the paradigm of selective motion estimation, a novel way of performing efficient motion estimation through reusing the information of surveillance video object tracker is proposed. In this approach, a real-time object tracker is used which generates information for each unique object with a unique track identity. In addition to this, objects are bounded in a rectangular box. So, instead of performing any kind of motion estimation for any block of the surveillance video, the motion vectors are calculated through the information generated by the object tracker.

Multiple reference frame based motion estimation increases the computational complexity with every extra reference frame. In order to address this problem, a fast full search for multiple reference frames based ME is proposed. This scheme is based on the successive elimination algorithm (SEA): a fast full search approach. This approach reduced the processing power for surveillance videos. Finally, another fast ME search algorithm, multi-pattern search algorithm, is proposed to find approximate calculations as in case of the well-known Diamond search.

1.3.3 Thesis Structure

Some background on the rate-distortion (RD) theory is presented in Chapter 2. In addition to the RD theory basics, some state-of-the-art strategies related to the surveillance videos are also presented. Chapter 3 describes the basic strategies for block-based video coding and wavelet-based scalable video coding. This chapter explains these techniques with the focus on the H.264/AVC standard which is the state-of-the-art block-based codec. Techniques of Motion Compensated Temporal Filtering (MCTF) and 2D DWT are explained for aceSVC: a wavelet-based video codec developed at QMUL. In chapter 4, object-based video coding implemented in MPEG-4 is presented. Object-based video coding has an attraction for surveillance videos because of independent handling of each object.

From Chapter 5 and onwards, research contributions for coding techniques specific to surveillance videos are discussed. Chapter 5 describes architectural modifications for a scalable video codec to convert it into the surveillance centric codec to improve the compression efficiency. A mathematical model of the modified architecture has been described. This chapter contains the experimental evaluation of the modified system as well as different experimental results for the road map towards a surveillance centric codec.

To overcome the problem of computational intensive motion estimation techniques, Chapter 6 describes a selective motion vector search technique based on the motion detection module. This technique maintains the visual quality of the video as that of full search yet improves the processing efficiency for the coding of the surveillance videos. A selective motion estimation approach has been proposed at different levels of selection (i) GOP level (ii) Frame level and (iii) Block level. All of these approaches are discussed with their challenges and experimental results.

Apart from selective motion estimation, a novel approach using a surveillance video object tracker will be discussed in Chapter 6. A unique motion track is calculated for each object of the surveillance video. The displacement of the object between the current and reference frame will be used to calculate motion vectors after identifying and matching the track in the two frames.

With the motive of maintaining the visual quality, Chapter 7 focuses on the fast full search approaches for efficient motion estimation. The concept of multiple reference

frames based motion estimation is discussed. A fast full search technique based on multiple reference frames is proposed. This technique improves the processing efficiency for surveillance videos while maintaining the visual quality.

Chapter 8 is related to efficient motion estimation. A multi-pattern based motion vector search technique has been proposed. The experimental evaluation of this approach is tested and compared against the popular diamond search and cross-diamond-hexagonal search techniques. This multi-pattern based search improves the processing efficiency at the cost of visual quality of the video. Finally, Chapter 9 gives the conclusions on the work presented in this thesis.

Chapter 2

Background

In this chapter, an overview of basics related to Rate Distortion (RD) optimisation and its application in video coding is presented. The fundamental measures related to RD optimisation are discussed. Some of the basic distortion measuring parameters are explained in the context of their use in video coding. The Lagrangion method for RD optimisation is presented because of its established useful application in video coding. Finally, some state-of-the-art techniques related to surveillance centric coding are discussed briefly.

2.1 Rate-Distortion Theory

Shannon with his rate-distortion theory [1] [2] addressed the elementary problem of RD optimization, maximizing reconstructed quality at the minimum cost (bit consumption). RD theory basically provides a way of minimising the number of bits representing the source element to a given reconstruction quality or distortion, which helps to develop a trade-off between rate and distortion.

As a branch of information theory [3]-[5], RD theory consists of fundamental concepts of information and entropy in information theory. A methodical explanation of all subsequent terminologies in information theory is beyond the scope of the work presented in this thesis; however a straightforward review of important concepts is presented in this preliminary section for the study of RD theory.

- **Entropy**

The quantitative evaluation of information in entropy is derived from Shannon's information theory [5]. Assume a random variable X takes values from the source elements $\{x_1, x_2, x_3, \ldots, x_{N-1}, x_N\}$ and $P(x_i)$ the probability that $X = x_i$, then entropy of source X is given by

$$H(X) = -\sum_{i=1}^{N} P(x_i)\log_b P(x_i) \qquad \textbf{(2.1)}$$

In this equation, the '$\log_b$' has the base of the working digital system. For digital binary system 'b' equals 2. The entropy of X, H(X), can further be considered as a quantitative representation in terms of the average number of binary symbols required to encode source X. As presented in the information theory, a lossless compression approach, the best compression performance in terms of preserving quality, can be accomplished by representing the source with the number of bits equal to its entropy.

- **Conditional entropy**

In case of lossy compression, some quantity of information is discarded during encoding process; consequently a reconstructed symbol will be different from the original source symbol. Therefore in RD theory a reconstruction element $\{y_1, y_2, y_3, \ldots, y_{M-1}, y_M\}$ is further dealt with, which is generally different from the source alphabet $\{x_1, x_2, x_3, \ldots, x_{N-1}, x_N\}$. To explain lossy compression, let us assume Y be a random variable that takes values from $\{y_1, y_2, y_3, \ldots, y_{M-1}, y_M\}$ and X from the source elements $\{x_1, x_2, x_3, \ldots, x_{N-1}, x_N\}$. Following equation (2.1), the entropy of the reconstruction is given by

$$H(Y) = -\sum_{j=1}^{M} P(y_j)\log_b P(y_j) \qquad \textbf{(2.2)}$$

Let $P(x_i \mid y_j)$ be a joint probability that $X = x_i$ and $Y = y_j$ occur with probabilities $P(x_i)$ and $P(y_j)$, respectively. Then the conditional probability is defined as

$$P(x_i \mid y_j) = P(x_i, y_j)/P(y_j) \qquad \textbf{(2.3)}$$

which presents a probability that $X = x_i$ occurs while the occurrence of $Y = y_j$ has been determined. Correspondingly the conditional entropy H(X | Y) is defined as

$$H(X|Y) = -\sum_{i=1}^{N} \sum_{j=1}^{M} P(x_i \mid y_j) \log_b P(x_i \mid y_j) \quad \textbf{(2.4)}$$

The conditional entropy H(X | Y) can be considered as the amount of uncertainty remaining about X given the knowledge of the value Y [5].

- **Average mutual information**

The average mutual information is another quantity measure that reflects the relationship between the uncertainty and entropy of two random variables [3]. It is defined as

$$I(X;Y) = \sum_{i=1}^{N} \sum_{j=1}^{M} P(x_i, y_j) \log_b [P(x_i \mid y_j) / P(x_i)] \quad \textbf{(2.5)}$$

By expanding the logarithm argument, I (X;Y) can be expressed by means of the entropy and the conditional entropy, as

$$I(X;Y) = H(X) - H(X|Y) \quad \textbf{(2.6)}$$

Thus, the average mutual information represents the amount of information that the knowledge of Y contributes to the reduction of uncertainty of X.

2.2 RD Optimisation for Video Coding

RD performance is the primary factor in the development and functioning of a video codec. Although it has been shown that video sources can be characterized by a Gaussian model [6], any statistical model has the possibility of failing in practical conditions due to the broad diversity of content in real world videos. Therefore for a particular video codec, the RD performance is usually evaluated in an operational approach, rather than attempting to capture accurately the characteristics of video sources using statistical models. Consequently, the goal of the RD optimisation for video coding is to search for the optimal operating point for a concrete system under a constraint, usually the bit-rate.

2.2.1 Distortion Measures

The method of measuring distortion is critical in RD optimization. A comprehensible distortion measure is not only vital for RD optimization, but also supportive for system design and performance evaluation. Relating to video coding, an appropriate distortion measure should be unswerving with the perceptual distortion perceived by the Human Visual System (HVS), since a human is ultimately to be the viewer. Subjective evaluation thus emerges as the most accurate approach for distortion measures in accordance with perception [7].

Regardless of the fact that objective video assessment may not match well with a subjective assessment founded on human visual perception, it is extensively used because of its simplicity and straightforwardness. The design of the objective distortion measure has concentrated on perceptual conformity, however, this aspiration becomes very vague as the characteristics of the HVS are complex and not well understood so far. For detailed information on perceptually compliant distortion measure, VQEG report [8] summarised the competitive proposals organized by video quality experts group, which has been devoted to the standardization of video quality assessment methods with a stress on objective distortion measurement. Concerning video coding, the most frequently used objective distortion measures in practice are the Sum of Absolute Differences (SAD), the Sum of Squared Differences (SSD), the Peak Signal-to-Noise Ratio (PSNR), and the Mean Squared Error (MSE). Their formulations are defined by

$$\mathrm{MSE} = (1/\mathrm{MN}) \sum_{x=1}^{N} \sum_{y=1}^{M} |f(x, y) - \tilde{f}(x, y)|^2$$

$$\mathrm{PSNR} = 10\log_{10}(255^2 / \mathrm{MSE})$$

$$\mathrm{SAD} = \sum_{x=1}^{N} \sum_{y=1}^{M} |f(x, y) - \tilde{f}(x, y)|$$

$$\mathrm{SSD} = \sum_{x=1}^{N} \sum_{y=1}^{M} |f(x, y) - \tilde{f}(x, y)|^2$$

Where f (x, y) and $\tilde{f}$ (x, y) are the pixel values at (x, y) in the original frame/block and the reconstructed frame/block, respectively, where M × N being as the frame/block size. The first two distortion measures are intuitively interpreted, where a lower MSE or SSD corresponds to lower distortion. Nonetheless, a higher distortion is related to a lower Signal-to-Noise Ratio (SNR) or PSNR. PSNR is widely used in evaluating compression performance in image or video coding, however, in motion compensated prediction, SAD and SSD are usually preferred on account of their succinct expressions.

2.2.3 Lagrangian Optimisation

Lagrangian optimisation is a standard method to solve the constrained optimization problem, which seeks to minimize an objective function subject to constraints on the probable values of the variable [10], [11]. In general, a constrained optimization problem can be described as follows.

Let S be a finite set of all permissible values of variable B. The objective function and the constrained function of B are denoted as O(B) and R(B) respectively, both of them real-valued functions of B defined for all $B \in S$. Then the constrained optimisation focuses on finding the best possible B given a constraint R_c, as

$$\min_{B \in S} O(B) \qquad \text{subject to} \quad R(B) < R_c \tag{2.7}$$

This constrained optimization problem can be solved by introducing a Lagrange multiplier λ and finding the solution to the corresponding unconstrained problem. It can be proved that for any $\lambda \geq 0$, the solution B *(λ) to the unconstrained problem

$$\min_{B \in S} \{O(B) + \lambda R(B)\} \tag{2.8}$$

is also the solution to the constrained problem posed by (2.7) with the constraint given by $R_c = R(B^*(\lambda))$ [10]. That is, for any $B \in S$ that satisfies the condition given by $R(B) \leq R(B^*(\lambda))$, the inequality $O(B^*(\lambda)) \leq O(B)$ holds.

According to theorem presented above, for a known non-negative λ, the best possible solution B*(λ) can be easily established by exploiting (2.8). The subsequent constraint $R_c = R(B^*(\lambda))$, consequently, varies with diverse selection of λ. However, in constrained optimization generally it is the constraint R_c, not the corresponding λ, which is predetermined before the optimization course of action. A problem then arises in how to find the appropriate λ in order to achieve the optimal solution B* (λ)

under a given constraint R_c. One may attain the Lagrange multiplier using bisection search [12], [13]. In RD optimization for video coding, however, a more computationally efficient approach is usually employed to determine the Lagrange multiplier [14].

Instead of solving the problem of Lagrange multiplier selection empirically as in [10]-[13], for RD optimized video coding the Lagrange multiplier can be determined for any unit as long as a quantization parameter is known, since λ can be formulated as a function of the quantization parameter [14]. Once the rate constraint R_c has been designated, the corresponding quantization parameters for each coding unit can be readily found through a rate control process such as in [15], and then the Lagrange multipliers in terms of a mapping to the quantization parameters.

As a distinct version of the general description given in (2.8), the constrained optimization problem in video coding can be solved under the motivation of Lagrangian optimization [6], [10]. The most favourable parameter set for all coding units can then be determined by minimizing the following cost function J

$$J = \sum_{i=1}^{N} d_{i,j} + \lambda \sum_{i=1}^{N} r_{i,j} \qquad \textbf{(2.9)}$$

given the Lagrange multiplier λ . Under the independent assumption that the rate $r_{i,j}$ and distortion $d_{i,j}$ can be measured independently for each coding unit, equation (2.9) can be rewritten as the sum of cost functions for all specific coding units:

$$J = \sum_{i=1}^{N} (d_{i,j} + \lambda r_{i,j}) \qquad \textbf{(2.10)}$$

$$J = \sum_{i=1}^{N} J_{i,j} \qquad \textbf{(2.11)}$$

Therefore the coding parameter for each coding unit can be optimized respectively by minimizing its own cost function, as

$$\min\{ J_{i,j} = d_{i,j} + \lambda r_{i,j} \}, \qquad \textbf{(2.12)}$$

for coding unit i = {1, 2, 3, ..., N} where j = {1, 2, 3, ..., M} for each i .

2.3 State-of-the-Art work

Object based techniques offered by MPEG-4 are exploited in [16]. Authors proposed a method to save storage space explicit to surveillance videos. Three different models

for background subtractions: a Mixtured Gaussian Model [17], a Non-parametric Background Model [18] and a Normalized Correlation Model [19], are discussed.

In the work presented in Vetro et al [16], a frame-based coding technique is utilised to compress a single background image, while the object-based coding technique is applied to compress the sequence of segmented foreground objects. A constant quality using fixed quantization parameters is used to code both background and foreground, and the background image is merely repeated for each reconstructed frame. In a real world system, it is anticipated that a predetermined criteria will be used to evaluate when the background image should be refreshed, if needed.

In the work of Du and Doermann [20], the approach presented by Vetro et al [16] was further extended by deriving the compression efficiency model that considers the number and size of foreground objects. This work was motivated by the observation that if the size and number of foreground objects is high, the object based coding may produce worse results than conventional frame-based coding. Therefore, by using the derived compression model the encoder can adaptively choose whether to perform frame-based or object based coding for a specific time segment of a surveillance video.

In order to get superior compression and better quality of video from a stationary camera, Nishi and Fujiyoshi [21] present a video coding technique based on pixel state analysis. Initially, pixel state analysis identifies foreground objects and the background using object detection. In addition to detecting the foreground objects, pixels of the foreground object are identified as stationary or transient. Despite the fact that the object is in motion, the foreground object has motionless pixels because it has same texture and colour. For motionless pixels, it is not required to store the pixel. It is feasible to reinstate the pixel of these stationary pixels by utilizing their values from the previous frame. For transient pixels, it is essential to store the pixel. Transient pixels of a foreground object are compressed by LZH (Lempel- Ziv-Huffman) codec which guarantees to restore the pixel's intensity entirely. When an object enters the surveillance area, object regions are identified as foreground and the other regions are identified as background. Foreground objects and background are encoded independently. At the stage of decoding, background and foreground objects are combined to restore the original image.

In the work presented by Cavallaro et al [22], the prefiltering step is used in effort to develop behaviour similar to the way humans treat visual information. An example is the division of the video into two groups of concern; explicitly foreground and background. The classification of the semantic division depends on the task to be carried out. Therefore, some a priori information of the objects to be segmented is necessary. For applications such as video conference or news broadcasting, faces may represent the semantic objects to be considered, whereas in applications such as video surveillance and sport broadcasting, motion information can be used as semantic for segmenting moving objects. The breakdown of the scene into significant objects prior to encoding is used in the perceptual prefiltering. Areas related to the foreground division, or semantic objects, are used as regions of interest.

Chapter 3

Block-Based Video Coding

Advance Video Coding (AVC) also known as H.264/MPEG-4 Part 10 is currently the most powerful and state-of-the-art video coding standard. It has been developed by a Joint Video Team (JVT) consisting of experts from the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) [24], [25], [26], [27].

In Figure 7, major functional blocks of AVC encoding process are shown [27]. An input frame F_k is presented for encoding. The frame is divided into units of 16x16 pixels called macroblock. Selecting either Inter or Intra Prediction mode, a prediction macroblock 'P' is formed using a reconstructed frame. In Intra mode, the samples of the current frame are utilised to construct a prediction 'P'. In the case of Inter mode, Motion Estimation and Motion compensation on one or more previously encoded and reconstructed frames are used to form 'P'.

.

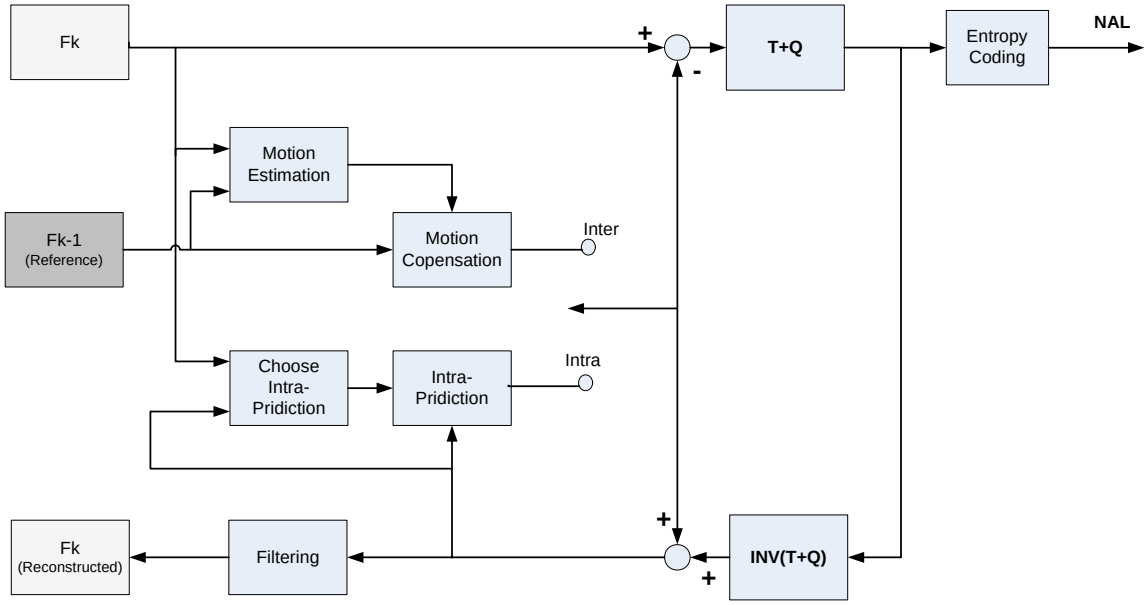


Figure 7: AVC Encoder (High Level View)

The difference between predicted macroblock 'P' and the current macroblock is transformed, quantized to produce 'X', a set of quantized coefficients. These coefficients are scanned for re-ordering and then entropy encoded. The compressed bit-stream consists of entropy-coded coefficients with the necessary information required to decode the macroblock (such as the macroblock prediction mode, quantization step size, motion vector information describing how the macroblock was motion-compensated, etc). This is passed to a Network Abstraction Layer (NAL) for transmission or storage.

The high level view of the decoder architecture is shown in Figure 8. The compressed bit-stream from the NAL is passed through the entropy decoding block then inverse transformation and inverse quantization processes are performed. After decoding the header of the compressed sequence, either Inter or Intra prediction mode is selected to produce the prediction signal. The prediction signal is added to the inverse transformed coefficients to reconstruct the frame. This reconstructed frame is passed through a filter to smooth the frame.

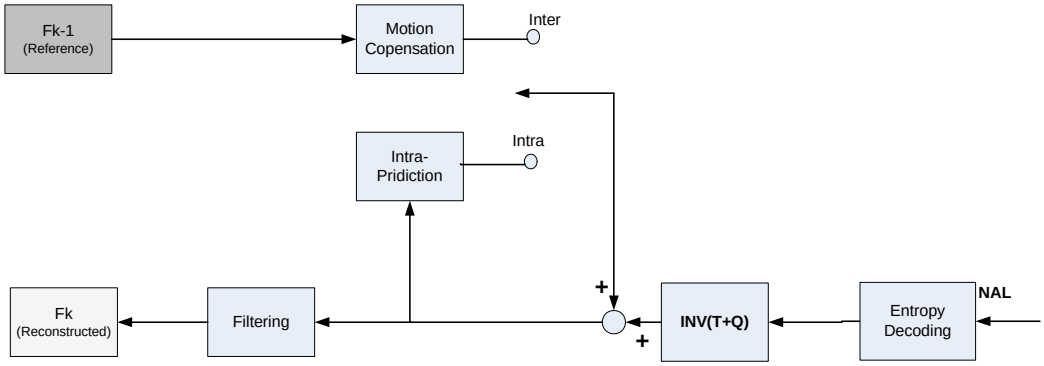


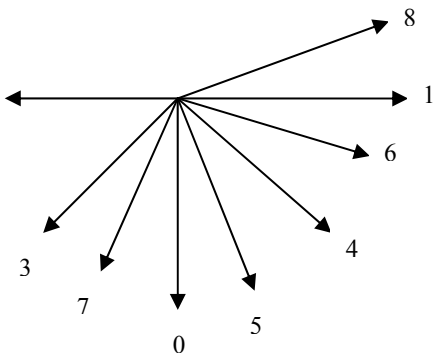
Figure 8: AVC Decoder (High Level View)

3.1 Intra Prediction Mode

In intra mode, previously encoded and reconstructed blocks of the current frame are used to form a prediction block. Then the difference between the current block and the prediction block is encoded. In AVC, a sub-block division of 4×4 is possible. For luma samples, the prediction block may be constructed for a 16×16 macroblock or for each 4×4 sub-block. AVC has 4 optional modes for a 16×16 luma block, 9 optional modes for 4×4 luma block and one mode for chroma block. The direction for prediction is given in Figure 9 and first four mode are explained by Figure 10

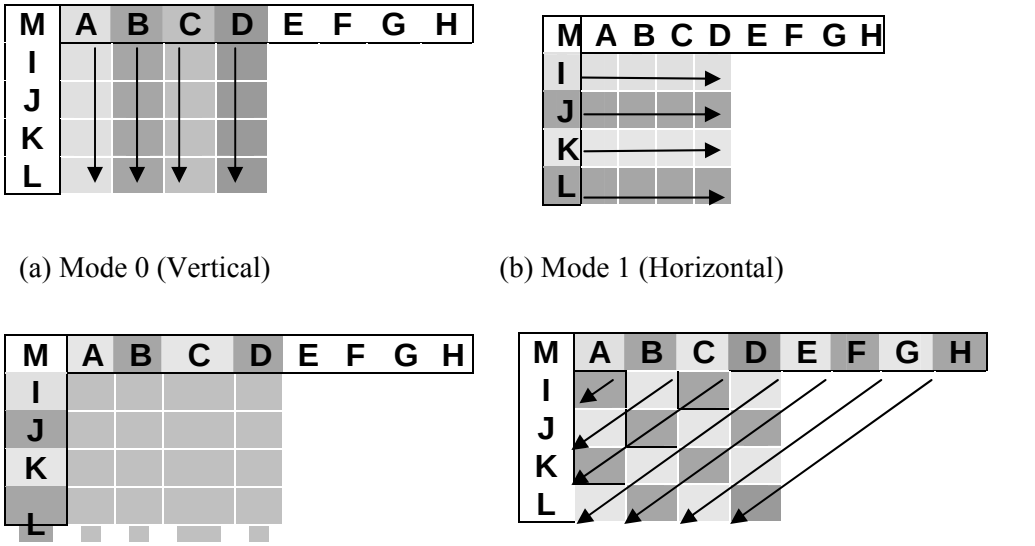
M	A	B	C	D	E	F	G	H
I	A	b	c	d				
J	E	f	g	h				
K	I	j	k	l				
L	M	n	o	p				

(a)



(b)

Figure 9: Spatial prediction directions (4×4 Block)



(a) Mode 0 (Vertical)

(b) Mode 1 (Horizontal)

(c) Mode 2 (Mean/DC)

(d) Mode 3 (Diagonal down-left)

Figure 10: Modes of 4×4

For 16×16 macroblock prediction, the pixel values of an entire macroblock of luma or chroma sample are predicted from the boundary pixels of neighbouring previously-decoded macroblocks. The encoder can select one of four different ways of performing full-macroblock prediction for each particular macroblock. These are: (i) vertical, (ii) horizontal, (iii) DC and (iv) planar. The pixel values of a macroblock are predicted from the pixel values just above or just left for vertical and horizontal prediction modes, respectively. In intra prediction mode 2 (DC prediction), the average of the luma values of the neighbouring pixels is used for the whole macroblock. In planar prediction, a curve-fitting equation with three parameters is used to form a prediction block. These parameters include brightness, slope in the horizontal direction, and slope in the vertical direction that approximately matches the neighbouring pixels.

Chroma intra prediction always operates using full-macroblock prediction. Because of differences in the size of the chroma arrays for the macroblock in different chroma formats (i.e., 8×8 chroma in 4:2:0 macroblocks, 8×16 chroma in 4:2:2 macroblocks, and 16×16 chroma in 4:4:4 macroblocks), chroma prediction is defined for three possible block sizes. The prediction type for the chroma is selected independently of the prediction type for the luma [27][28].

In 4×4 prediction modes, the values of each 4×4 block of luma data are predicted from the adjacent pixels above or left of a 4×4 block, and encoder can select nine different directional ways of performing the prediction.

3.2 Inter-Prediction Mode

The spatial redundancy present in the video is handled through intra prediction modes. To exploit the temporal redundancy present in the video, inter prediction mode is used. At the core of inter prediction mode are techniques of motion estimation and motion compensation. AVC supports a wide range of block size (down to 4×4) and fine sub-pixel motion vectors.

The luminance component of each macroblock may be divided in four different ways as shown in Figure 11 [27]. If 8×8 encoding mode is chosen then each of the four 8×8 macroblock partitions may further be split in sub-partitions in 4 different ways as shown in Figure 11. So, within each macroblock, there are a large number of possible combinations of partitions and sub-partitions. This approach of dividing macroblock into partitions of varying sizes is known as tree structure motion compensation.

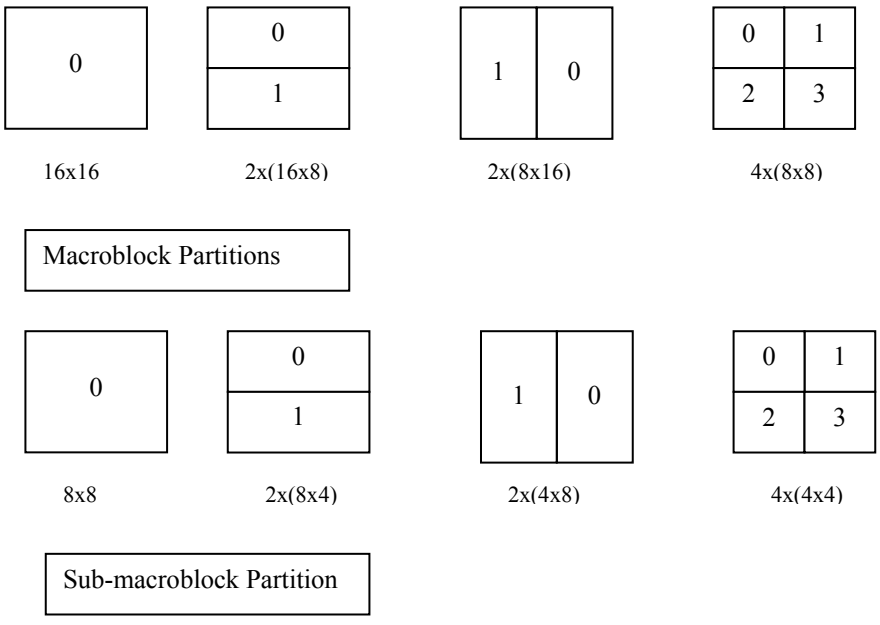
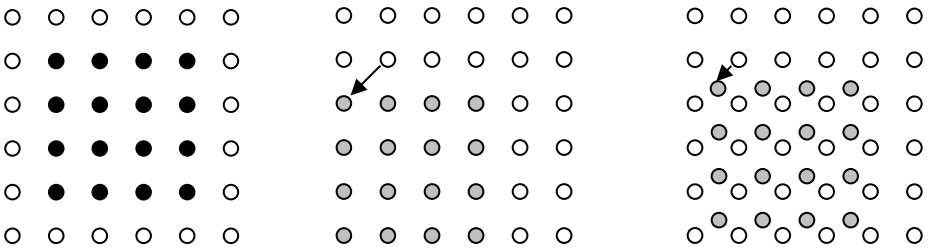


Figure 11: Macro and Sub-macroblock partitions

A separate motion vector is associated with each partition or sub-partition. The choice of partition and the associated motion vector is encoded and included in the compressed bit-stream. Therefore, the coding efficiency decreases with the high number of partitions (smaller blocks). On the other hand, a small number of bits are required for large sized partitions; however the motion compensated residual may contain a significant amount of energy in frame areas with high detail. Generally, homogeneous areas of the frame are divided into large partitions while a small partition size may be more useful for detailed areas.

Each partition in an inter-coded macroblock is predicted from an area of the same size in a reference picture. The motion vector represents the offset between the two areas. For luma components, the AVC supports quarter pixel resolution. If sub-pixel position in the reference frame does not have the pixel then it is created using interpolation from the neighbouring image pixels. In Figure 12, the integer pixel and sub-pixel prediction are shown [27]. Sub-pixel motion compensation increases the complexity of the system but it offers significantly better compression efficiency than integer-pixel motion compensation.



a) Current Frame (b) Reference Block: MV (-1,1) (c) Reference Block: MV (-0.5, 0.75)

Figure 12: Integer pixel and sub-pixel motion estimation

3.3 Transform and Quantisation

Each residual macroblock is transformed to decorrelate the data spatially. Depending on the type of residual data, there are three transforms for the baseline profile of AVC: a transform for the 4x4 array of luma DC coefficients in intra macroblocks (predicted in 16×16 mode), a transform for the 2×2 array of chroma DC coefficients (in any macroblock) and a transform for all other 4×4 blocks in the residual data.

Further transforms are chosen depending on the motion compensation block size if the optional “adaptive block size transform” mode is used.

The transformation step produces coefficients which are quantized using a quantization control parameter. This parameter can be changed for every macroblock. When the video format supports 8 bits per sample then the quantization parameter can have one of the 52 possible values. If the video format supports more than 8 bits per sample then the fidelity range extension (FRExt) expands the number of steps by 6 for each additional bit of sample. The relationship between the quantisation step and quantisation parameter is not linear. An increment of 6 in the value of quantisation parameter doubles the quantisation step size.

3.4 Entropy Coding

The term Entropy Coding refers to lossless coding techniques that exchange data elements with coded representations. These techniques can result in a significantly reduced data size. In H.264, two modes of entropy coding are used: binary arithmetic coding (BAC) and variable length coding (VLC). Both of these techniques are applied in a context adaptive (CA) way, resulting in the terms CAVLC and CABAC.

codeNum	Code
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000
---	---

Table 1: Exponential Golomb Code (UVLC)

The main idea of VLC is that when the data elements to be coded occur with different frequencies; the data elements with high frequency of occurrence can be assigned short codes, while the data elements with low frequency of occurrence can be assigned longer codes. This results in variable length codes to data elements. For the syntax elements other than residual transform coefficients; Universal Variable Length

Coding (UVLC) is applied. First eight elements of the Exp-Golomb Code table for given input data elements are shown in Table 1.

The codeNum is used as an index to the actual data elements. The general form of these codes is [‘n’ zeros][1][n-bit DATA]; such that DATA is a binary representation of an unsigned integer. Such a code is decoded as

$$\mathbf{codeNum = 2^{n} + Int(DATA) - 1}$$

such that the Integer (DATA) is the integer corresponding to the binary code [28].

The AVC standard specifies twelve additional code tables to achieve higher efficiency through coding the abundant residual transform coefficient. Six tables for characterizing the content of the transform block as a whole, four for indicating the number of coefficients, one for indicating the overall magnitude of a quantized coefficient value, and one for representing consecutive runs of zero-valued quantized coefficients. The length of the fixed length coefficient value suffix and the selection of the appropriate table are based on the statistical characteristics of the current stream; thus the context based (CAVLC). The coding efficiency is increased in CAVLC at the cost of execution efficiency.

Further improvement in the coding efficiency can be achieved using context-based adaptive binary arithmetic coding (CABAC) at the cost of increased complexity. The CABAC based coding has roughly 10% higher coding efficiency than CAVLC [28]. In CABAC, non-integer number of bits per symbol can be assigned; offering a very high degree of statistical adaptivity which allows the coder to adjust to changing symbol statistics, and context selection ensures that the statistical adaptivity is relevant to the specific data being coded [28]. In the case of CAVLC, only the transform coefficients are coded adaptively whereas in CABAC, intra prediction modes, the macroblock type, motion vectors, residual transform coefficients and several other syntax elements are coded adaptively.

3.5 Scalable Video Coding

Generally, a scalable video bit-stream is composed of such sub-streams which can be removed from the main bit-stream to form a valid bit-stream for reconstructing the source video data but with lower quality as compared to the complete original bit-stream. The usual modes of scalability are spatial, temporal and quality scalability. A

spatially scalable bit-stream consists of sub-streams which represent the reduced spatial dimensions (picture size). In temporal scalability, the sub-stream represents the source video with a reduced frame rate. With quality scalability, the sequence reconstructed from the sub-stream has the same spatial and temporal level as with full bit-stream but offers lower visual (perceptual) quality. Scalable video coding (SVC) offers a number of benefits in terms of applications [29], [29]. A good overview of scalable extension of H.264/AVC is presented in Schwarz et al [31].

3.6 Wavelet-Based Scalable Video Coding

In surveillance videos [29], [32], the scalable video codec (SVC) can be used to adapt to the appropriate scalability level when an event of interest is detected. The SVC has to be modified for the requirements of surveillance applications [33]. In this section, the wavelet-based SVC ([34], [35]) framework is presented which is named as aceSVC. In this framework, the temporal and spatial scalability has been achieved through the process of motion compensated temporal filtering (MCTF) [36] in the temporal domain and 2D discrete wavelet transform (DWT) ([37], [38]) in the spatial domain, respectively. The MCTF results in motion information and wavelet coefficients that represent the texture of transformed frames. These wavelet coefficients are then bit-plane encoded [39] to achieve quality scalability. The architecture of the aceSVC encoder is shown in Figure 13.

The architecture is divided into Spatio-temporal decompositions, Rate-distortion optimization and Entropy coding. The spatio-temporal decompositions section enables signal decorrelation suitable for compression of video. In wavelet video coding, the spatial decomposition of original frames or motion compensated frames is achieved through spatial discrete wavelet transform (2D DWT). The temporal transform is performed using the concept of motion compensated temporal filtering (MCTF). MCTF can be applied on the original frame or on the spatial sub-bands. The motion information obtained from the motion estimation block is also used in the process of MCTF. On the basis of order of applying spatial and temporal transform, aceSVC architecture supports two schemes:

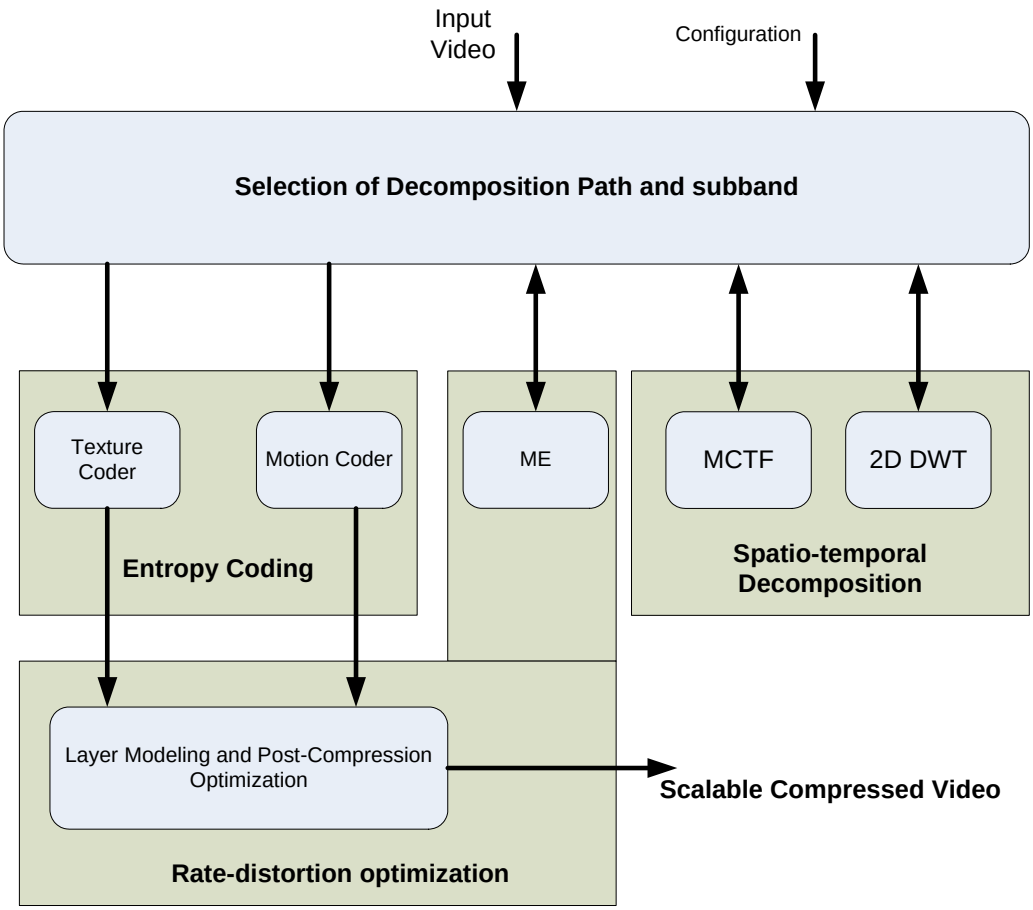


Figure 13: aceSVC Encoder building blocks

2D + t : spatial transform followed by temporal transform

t + 2D : temporal transform followed by spatial transform

The order of applying the decomposition steps defines a decomposition path, which in the aceSVC encoder is created according to the input target points [40]. The sub-band selector inputs the resulting sub-bands to the standard video compression modules like quantization and entropy coding. The remaining redundancy present in the data of motion information and quantized wavelet sub-band coefficients is removed through entropy coding. Using the bit-stream organization methods developed for scalable video coding [41], the compressed data is organized into the compressed bit-stream.

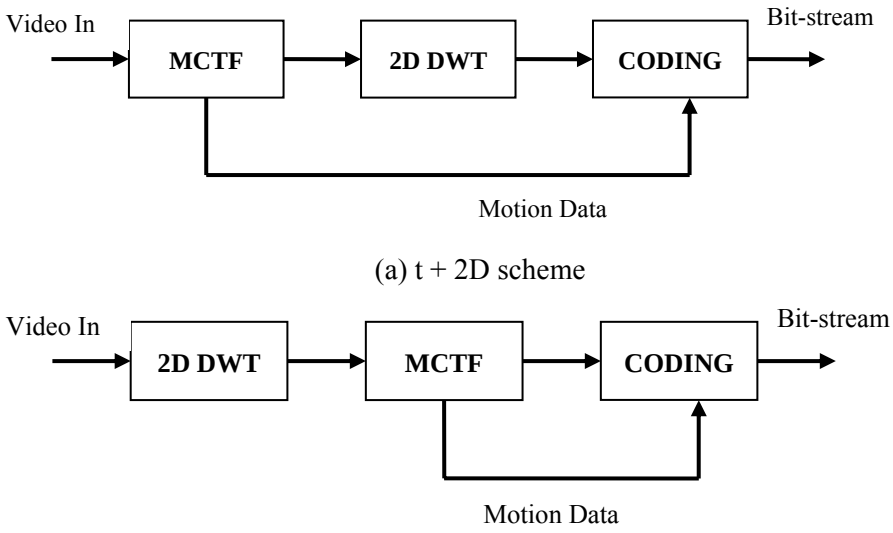


Figure 14: Decomposition schemes

3.7 Temporal Transform

Proper exploitation of temporal redundancies can result in a great efficiency in the video compression. A frame can be represented with data from already coded frames because they contain similar information. A significant part of temporal redundancy can be removed through the process of motion compensation. In this process, a predicted frame is divided into motion units, each of which is associated with a motion vector and motion mode for the description of prediction. Motion models are used to define a prediction method (Inter/Intra) for each motion unit, and the relevant motion unit configurations. Motion vectors are associated with the motion units in temporal prediction (inter mode) to describe the displacement of motion units between frames.

Temporal filtering or decomposition has been adopted as a generalized approach to achieve temporal scalability and motion compensation. If we consider the dyadic decomposition, then the even numbered frames become low-pass frames (L frames), while the odd number frames become high-pass frames (H frame). The low-pass frames can be used for further decomposition. This process is known as MCTF because of filtering along the motion trajectories. Initially, MCTF has been developed for full frame resolution [36], [42], [43]. To enhance the coding performance at lower

spatial resolutions, different approaches of applying MCTF to different spatial sub-bands of the input frames has been developed [35], [45], [46], [47].

In full frame MCTF, the iterative filtering of the low-pass frames from the previous temporal level can result in higher depth of temporal decomposition. The application of the Haar wavelet filter (filtering between two frames only) for performing MCTF is shown in Figure 15.

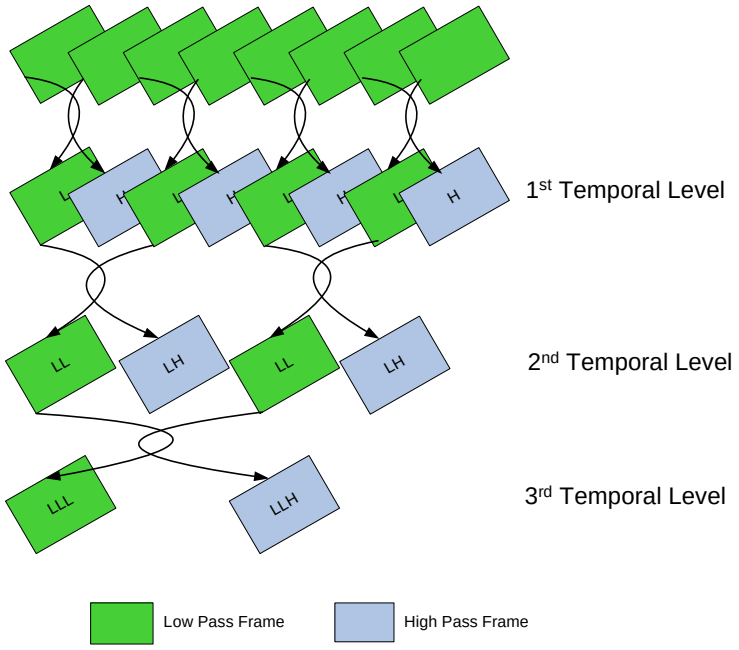


Figure 15 Motion compensated temporal filtering

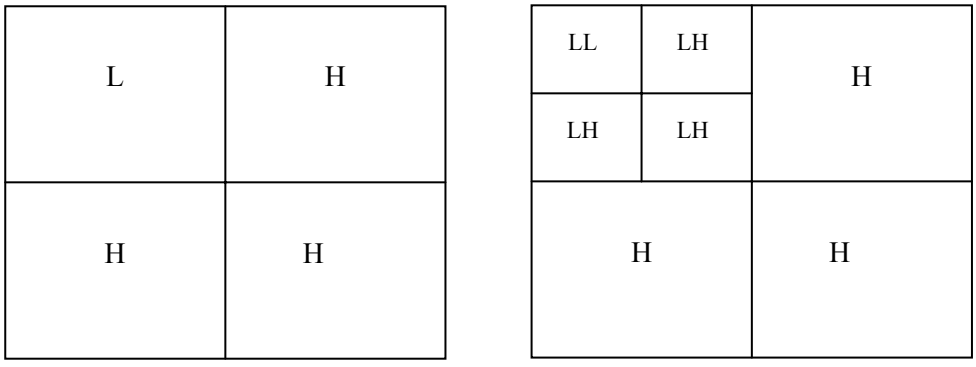
In this particular example, three levels of MCTF are performed on the original frames, resulting in four temporal sub-bands. After three time low-pass filtering, the lowest sub-band and its corresponding frames are labelled with LLL. The corresponding high-pass sub-bands are shown with labels H, LH, LLH, respectively for consecutive lower temporal levels. The number of performed low-pass filtering steps necessary to obtain a specific sub-band is denoted by the number of "L" letters. In the context of temporal scalability, a basis is obtained for embedded encoding of "T + 1" layers of different frame rates by performing filtering of "T" temporal decomposition levels. In each step of MCTF, the motion information is produced which is together with the resulting frames is used for decoding the frame. The video synthesis process starts with the lowest sub-band LLL and LLH and follows the inverse MCTF stages from

the lowest temporal level to the highest levels. In the synthesis of each temporal level, the corresponding motion information is used.

3.8 Spatial Transform

To exploit the spatial redundancies, there are two major categories of spatial transforms utilized in the coding of video and still image- discrete wavelet transform and Discrete Cosine Transform (DCT) or DCT-like transform [48]. In Joint Picture Expert Group (JPEG) still image coding standard [49], DCT is used which is also part of several video coding standards.

Discrete Wavelet Transform (DWT) is used in the state-of-the-art JPEG-2000 still image coding standard [50], [51]. The wavelet transform offers good compression efficiency as well as inherit feature of scalability. For this reason, a number of scalable video codecs has adopted wavelet transform. In most of the proposed solutions, the scalable video coding architecture has been divided into two categories in regard to the order of dimensions in which the transform is performed. In 't + 2D' scheme, spatial wavelet transform is performed after the temporal decomposition, Figure 16(a). In this approach, motion compensated temporal filtering (MCTF) is performed on the full spatial resolution as opposed to the wavelet sub-band domain. This scheme is known as spatial domain MCTF (SD-MCTF). Several codecs has been developed which are based on the SD-MCTF [43], [52], [53].



a) First Level b) Second Level

Figure 16: Spatial Decomposition

SD MCTF frameworks represent texture energy very compactly. As the motion information is generated from the application of MCTF on the full frame resolution, it

causes the problem of scaling the motion information at lower resolutions. It means that lower spatial, temporal and quality scale streams utilize motion information which does not necessarily represent the optimal block-matching decision for these particular streams [40]. Alternative approaches have been developed to address this problem. In one of the approaches, input frames are spatially transformed and then the transformed frames are used for temporal decomposition [45]. This approach is known as in-band MCTF (2D + t). Since motion compensation is performed separately for different spatial resolutions, in-band MCTF offers a better level of scalability in video [40]. Each spatial decomposition has its own motion information which can be used to optimise the rate-distortion for different decoding points. On the other hand, this approach adds extra complexity due to the fact that it performs motion estimation and motion compensation at different resolutions. Further, the MCTF on the full spatial domain is more efficient as compared to the MCTF on spatial sub-bands.

3.9 Scalability support in bit-stream

In aceSVC, the compressed bit-stream is organized to support the extraction of all types of scalability features. The bit-stream of the video sequence is divided into group of pictures (GOP). Each GOP consists of GOP header with spatially and temporally decomposed frames.

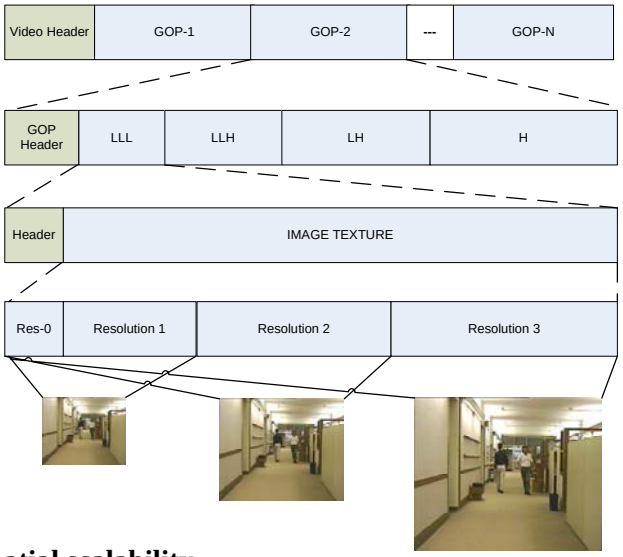


Figure 17: Spatial scalability

For spatial domain MCTF, the low-pass temporal frames contain the texture data in different spatial resolutions. In Figure 17, three different spatial resolution levels in the bit-stream are shown. The bit-stream of each resolution is organized in a layered structure. The lowest resolution can be obtained by extracting only the base layer. Higher spatial resolutions can be decoded by extracting the bit-stream of base layer with additional layers.

The quality scalability also known as SNR scalability can be characterized as offering the same video sequence with different quantization levels. Generally, the extraction of a larger portion of bit-stream results into better decoded visual quality. In Coarse Grain Scalability (CGS), sub-band coefficients are organized in layers. This helps efficient adaptation and use of different tools as presented in [54]. Three different quality levels are shown in the bit-stream of Figure 18. The lowest quality sequence is decoded by extracting the Q0 bit-stream. Further quality may be improved by adding the extra layers.

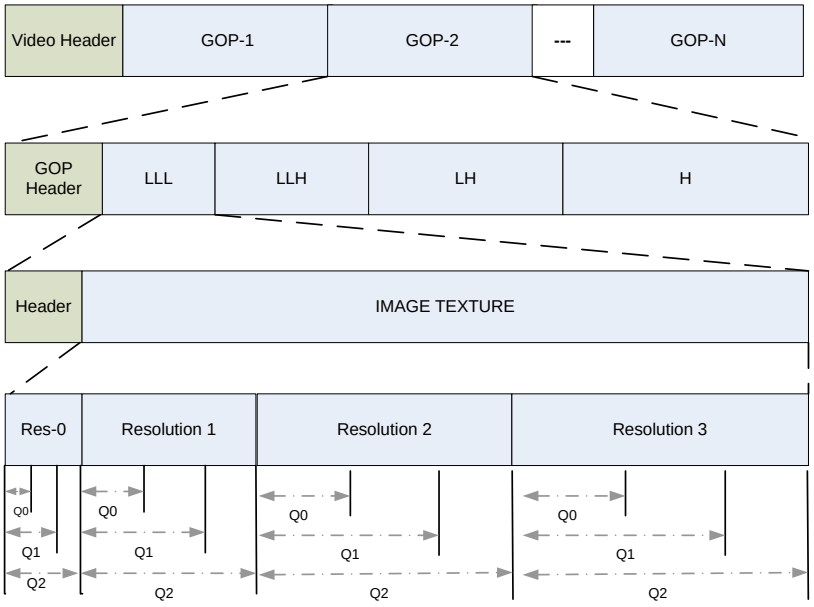


Figure 18: Quality scalability

Temporal scalability is to offer the same sequence with different frame rates. The process of MCTF generates low-pass and high-pass frames. Like low-pass frames, high-pass temporal frames are also decomposed spatially so as to maintain the correspondence with its low-pass frames. The bit-stream of high pass texture data always contains the associated motion information required for its reconstruction.

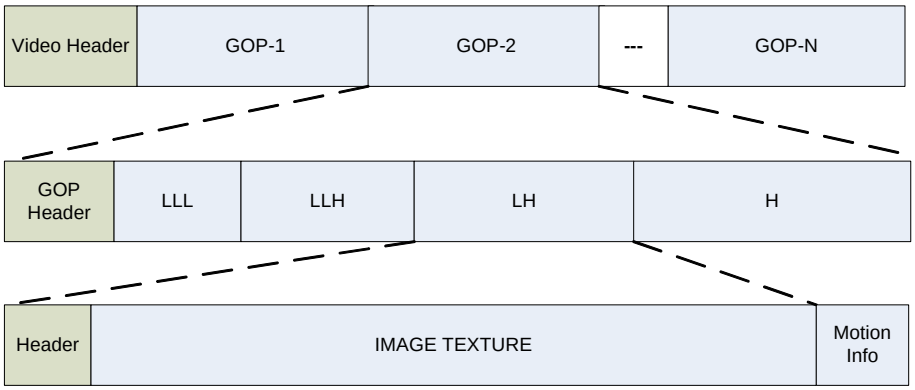


Figure 19: Temporal scalability

Chapter 4

Object-Based Video Coding

Replacing the conventional rectangular block based video coding with the object-based coding is the main concept defined by the MPEG-4 standard. Object-based video coding gives direct access to the scene contents; thus helps manipulation of each object independently. According to MPEG-4 requirements [55], object-based video coding shall provide the following features [56]:

- Object-based representation
- Object quality and fidelity
- Object-based bit-stream manipulation and editing
- Object-based coding flexibility
- Object-based random access

Each object described in the video is characterized by spatial and temporal information in the form of shape, motion and texture [57]. Following are the basic definitions for the hierarchical structure of MPEG-4 object based coding:

♦ **Visual Object Sequence (VS):** The complete MPEG-4 scene which may contain any natural or synthetic objects and their enhancement layers [57].

- **Video Object (VO):** An area of video sequence (VS) that may have arbitrary shape in the 2D domain and may exist for an arbitrary length of time.
- **Video Object Plane (VOP):** A video object (VO) in a particular instant of time is termed as video object plane.
- **Group of Video Object Planes (GOV):** GOV provides random access points into the compressed bit-stream. At these access points, VOPs are encoded independent of each other [57].

In Figure 20, three different VOPs are presented: VOP1: person facing the camera, VOP2: the person with his back towards the camera and the VOP3: the whole background. If a visual scene contains a synthetic object then the shape and characteristics of the object already exist. For a natural object present in the scene, a process of segmentation has to carry out. A wide range of segmentation techniques have been developed. These techniques can be broadly divided into three categories:

- Pixel-Based Segmentation
- Edge-Based Segmentation
- Region-Based Segmentation

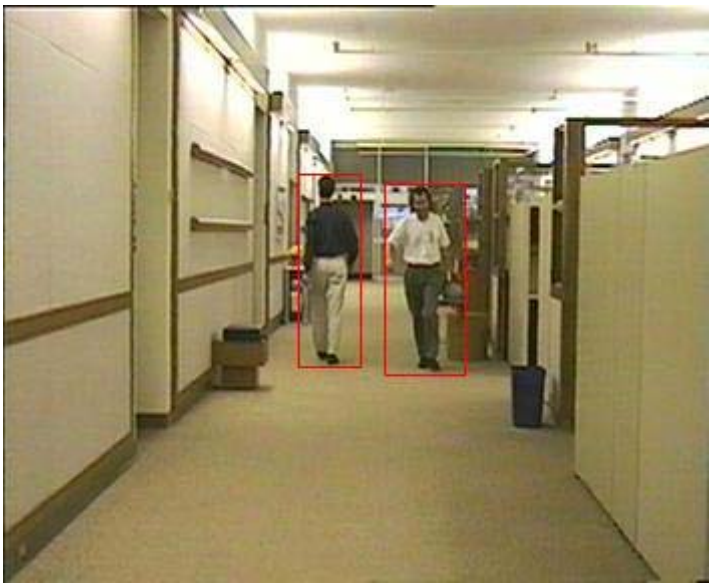


Figure 20: Different VOPs in a frame

4.1 Object Shape Coding

After detecting the object by performing the segmentation step or any other analysis technique, the shape information of each object is described by a binary alpha plane or greyscale alpha plane. In binary alpha plane, each pixel is tested if it is an inside shape (1) or an outside shape (0). On the other hand in grey scale plane, the shape information is represented with 8 bits.

4.1.1 Binary Shape Coding

In a bitmap scheme, a binary shape is represented in a matrix of binary values. The bitmap based techniques are very simple and have low computational complexity with good compression results.

A motion compensated block based technique is applied to binary shape coding. As a first step of binary shape coding, a rectangular window with a size of a multiple of 16 pixels in horizontal and vertical directions is selected such that this window can bound the shape of the object. The position and size of this rectangular windows is chosen such as to minimize the number of '16×16' sized non-transparent blocks. Samples in the bounding box and outside the VOP are set to 0. The encoding process is performed block by block after partitioning the bounding rectangular box into blocks of '16×16' pixels.

Matrix representation of the object shape is referred to as a binary mask. Pixel inside the VOP shape is set to 1 while the pixel outside the shape is set to 0. It is then partitioned into '16×16' blocks referred to as binary alpha blocks (BAB). Each BAB is encoded separately. Most of the BABs have pixels all of the same value, either 1 (opaque block) or 0 (transparent block). A Context based Arithmetic Encoding (CAE) and motion compensation techniques are applied to the encoding of BAB. The terms of IntraCAE and InterCAE are used for without or with motion compensated CAE, respectively. The motion vectors computed in the InterCAE are differentially coded. The combination of CAE and motion compensation techniques helps to result in the following mode for BAB encoding [57]:

1. The block is flagged opaque. In this case, shape coding is not necessary but texture information has to be coded.
2. The block is flagged as transparent. No shape and texture coding.
3. The block is coded using IntraCAE without use of past information.
4. Motion vector difference (MVD) is zero but the block is not updated.
5. MVD is zero and the block is updated. InterCAE is used for coding the block update.
6. MVD is non-zero, but the block is not coded.
7. MVD is non-zero, and the block is coded.

4.1.2 Greyscale Shape Coding

The binary shape and greyscale shape corresponds to the same structure but greyscale shape pixels can have any value from (0-255) thus indicating the level of transparency/opaqueness. The pixel with value 0 is considered to be completely transparent and a pixel with the value 255 is considered to be completely opaque. The pixel value between 0 and 255 corresponds to degree of transparency in the pixel. A block based motion compensated DCT is used for the encoding of greyscale shape information.

4.2 Foreground Coding

Foreground objects are coded using conventional techniques tailored for the arbitrary shape of the object. Foreground coding is divided into two major steps: (i) Motion estimation and compensation and (ii) Texture coding.

4.2.1 Motion Estimation and Compensation

The object based coding in MPEG-4 uses the same techniques as for block based coding to exploit the temporal redundancies. These techniques have been adapted for object based coding. There are three modes for encoding VOP in MPEG-4 [57]:

1. Intra VOP (I-VOP): a VOP encoded independently of any other VOP.
2. Predicted VOP (P-VOP): a VOP may be predicted based on another previously decoded VOP using motion compensation.

3. Bidirectional Interpolated VOP (B-VOP): a VOP may be predicted based on the past as well as future VOPs. The B-VOP cannot be interpolated using another B-VOP.

Macroblocks in the bounding rectangular box are utilized for motion estimation. Motion estimation is performed in the usual way if the macroblock is completely within a VOP. For macroblocks that partially belong to the VOP, motion estimation is performed using modified block matching approach. Sum of Absolute Difference (SAD) is computed only for those pixels in the macroblock which belong to the VOP [57]. For the VOP boundary reference blocks, the pixels outside the VOP are assigned values through padding. For such blocks, the SAD is computed including padded pixels.

4.2.2 Boundary Macroblocks Padding

To pad the boundary macroblocks, horizontal and vertical extrapolation is performed. In the first step, the pixel extrapolation is performed in a row. If only one end of the row has opaque pixels then all the transparent pixels in the row are replaced with the value of the nearest opaque pixel of the row. If the both ends of the row have opaque pixels then the transparent pixels in the row are replaced with the value equal to the mean of the two neighbouring pixels. After padding the row horizontally, the same process is repeated for columns in the vertical direction.

4.2.3 Exterior Macroblocks Padding

There is a possibility that the referred macroblock may fall entirely outside the VOP. Such macroblocks are completely transparent. For the padding of such macroblocks, the neighbouring macroblocks are utilized. In Figure 21, the order of neighbouring macroblock selection is shown. If MB1 is a boundary MB of the object, then it is selected for performing padding. If MB1 is not a boundary MB, then MB2, MB3, MB4 are tested sequentially for boundary MB. If no boundary MB is found then pixels are filled with the value of "2^n - 1" with 'n' being the number of bits per pixel.

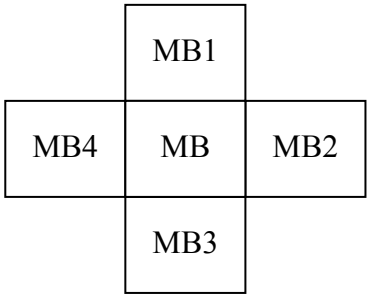


Figure 21: Order of MB selection for performing padding

4.2.4 Texture Coding

In the object based coding, if the block being encoded has all opaque pixels then it is encoded with conventional 8×8 block DCT. In the case of boundary blocks which have both opaque and transparent pixels, either DCT is applied after padding the blocks or a Shape Adaptive DCT is applied. For a normal DCT approach, for an N×N block, there are N^2 coefficients. The shape adaptive DCT is more efficient because it uses only opaque pixels.

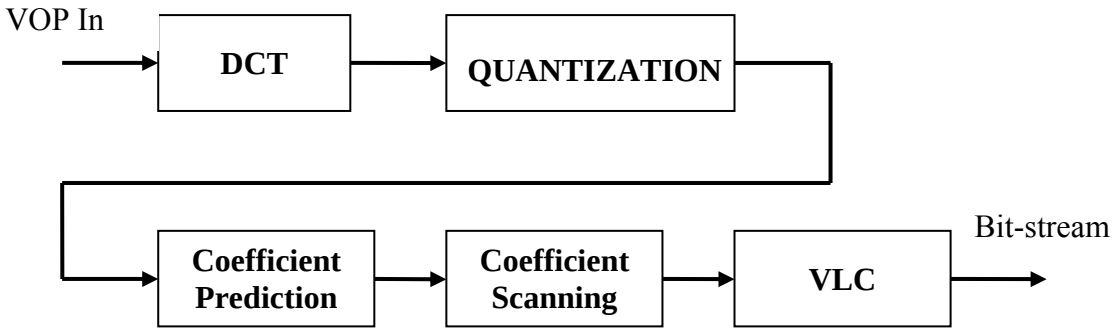
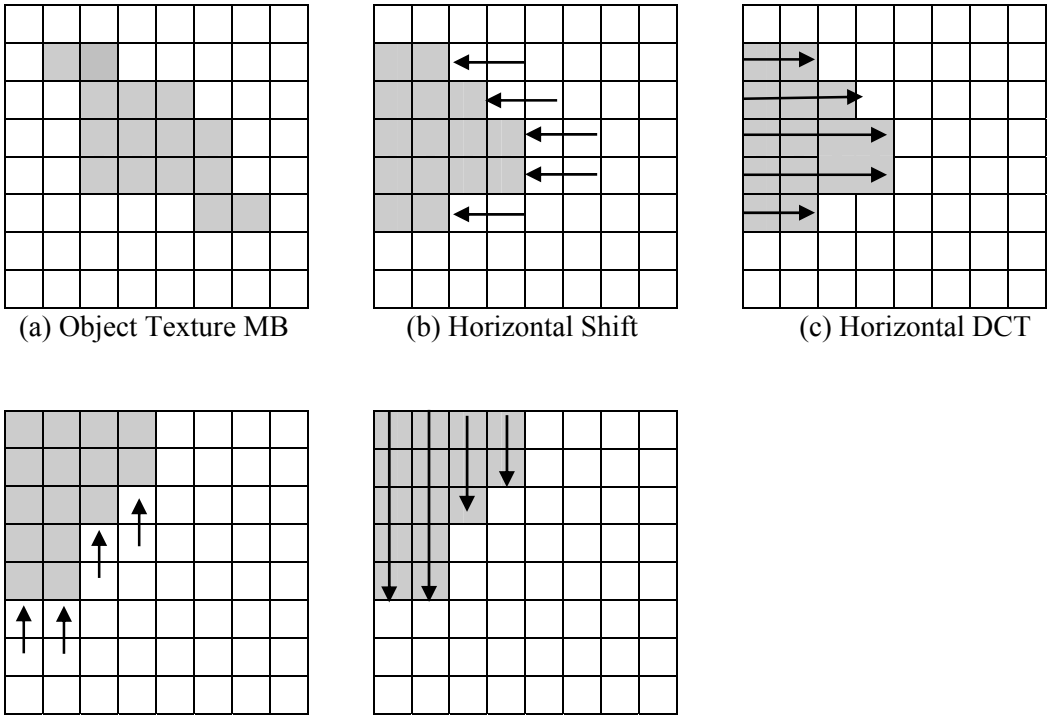


Figure 22: Blocks for encoding VOP texture

- ***Shape Adaptive DCT (SA-DCT)***

In shape adaptive DCT, each VOP is processed in 8×8 blocks. In each block, first, each row is processed. In each row, all the opaque pixels are shifted to the left such that the left most opaque pixel touches the left boundary of the block. Then a one dimensional DCT in the horizontal direction is performed to transform the pixels. After processing all the rows, then the resulting block with coefficients is processed for each column shifting all the non-zero coefficients to the top of the block. Then on the shifted column, again one dimensional DCT is performed in the vertical direction.

This is shown in Figure 23. For intra coding of VOPs, an additional step of computing the zero-mean block is carried out.



(a) Object Texture MB (b) Horizontal Shift (c) Horizontal DCT

(d) Vertical Shift (e) Vertical DCT

Figure 23: Shape adaptive DCT process

4.3 Background Coding

Compression efficiency of object based coding is higher because of the isolated treatment of background. Using only background, it can be coded with lower quality and lower reliability because of being visually a less important portion of the image. In addition, in most of the cases, the background does not have much change thus causing the improvement in the compression efficiency. In one method for background coding, first a low quality approximation of the background is sent then afterwards, progressively, some areas of this background with higher quality are sent.

Chapter 5

Surveillance Centric Coding

In this chapter, we introduce the coding paradigm of Surveillance Centric Coding (SCC), in which coding for specific surveillance applications is targeted. The SCC aims at exploiting specific properties of surveillance video in a comprehensive application framework including coding adaptation to surveillance, rate distortion optimisation according to the VCA (video content analysis), and other related concepts.

The basic approach towards Surveillance Centric Coding (SCC) has been introduced. The Scalable video coding (SVC) has attraction for surveillance applications because of its spatial, temporal and quality adaptability. In this chapter a scalable video codec, aceSVC (discussed in chapter 3), has been modified to support event-based video coding of the surveillance videos. The architectural modifications for GOP level switching of the scalability features have been presented and implemented. This approach enables the saving of bit-rate and storage space. The following sections give a description of the modified system.

5.1 Architectural Modifications

The required information in the surveillance video is the presence of some motion activity or any other event of interest. Therefore, the portion of the video which has an event of interest should have a good RD performance. On the other hand, the portion of the video which does not have any special information may be set to a lower RD performance. This adaptive RD performance saves a large number of bits. On the other hand, the bit-stream generated by the scalable video codec does not support GOP level switching of the scalability features. Thus to support event based scalability features at GOP level for the coding of surveillance videos, the architecture of aceSVC has been modified. The adaptive RD performance is achieved at the GOP level through the modified architecture. Some new blocks have been introduced to implement this approach. These blocks are:

1. GOP Selector
2. GOP Collector
3. GOP Analysis

5.1.1 GOP Selector

The GOP selector (GS) block has been used in all three stages of scalable video coding (Encoding, Extraction and Decoding). At each stage, it manages one GOP from the input sequence and forwards it to the following blocks for processing. The use of this block removes the dependency of the GOP on the following and preceding GOPs; and enables extractor to extract each GOP with different spatial, temporal and quality level.

5.1.2 GOP Collector

Like GOP selector (GS) block, the GOP collector (GC) block is part of each stage of scalable video coding. Its function is to receive the processed GOP and to manage it for the final output bit-stream. At the encoding stage, it also accepts GOP analysis information and associates it as metadata with the processed GOP.

5.1.3 GOP Analysis

This block finds out if a certain event of interest is present in the selected GOP. The main event of interest in the surveillance video is the motion of the object of interest. This block analyzes each GOP to detect the motion present in the GOP. The information produced by this block is used to extract the appropriate spatial, temporal and quality features.

The uncompressed raw video can be represented with equation (5.1), where G_i is the *i*th uncompressed GOP. Each GOP can be represented with its frames (f_j) with equation (5.2).

$$V_{RAW} = \{ G_1 , G_2 , G_3 , \ldots, G_N \}. \quad \textbf{(5.1)}$$

$$G_i = \{ f_1 , f_2 , f_3 , \ldots, f_M \} \quad \textbf{(5.2)}$$

where,

N = Total number of GOPs in the video

M = Total number of frames in a GOP

The algorithm for analysis is shown in Figure 24. The GOP analysis block is capable of detecting multiple motion levels in the video. This helps to produce a bit-stream with multiple scalability features in different GOPs for a single video derived by the analysis result.

At the start, f_1 is set as the reference frame and the following frames are compared with it to find out the similarity of frames with f_1 . If the similarity is below a certain threshold level ' p_1 ', then GOP is considered to have a moving object with motion level 1; and if the similarity is below threshold ' p_2 ', then motion level 2 is assigned. Thus multiple motion levels have been defined.

The algorithm of GOP analysis always looks for the maximum motion level in the sequence. Once the maximum motion level is detected then it terminates to save time and processing power. Selection of threshold values is the key to the accuracy of analysis. Based on the analysis of the GOP, layer extraction index 'μ' is generated, which is used by the extractor to extract an appropriate scalability layer.

```
1-  MOTIONLEVEL = 0
2-  THRESHOLD1 = p1
3-  THRESHOLD2 = p2                    comment: p1 > p2
4-  Ref_Frame = first frame of GOP
5-  FOR each frame of GOP except first
6-    similarElements = 0
7-    FOR each element of the frame
8-      IF element is equal to related element in Ref_Frame
    THEN
9-        Increment in similarElements;  END IF
10-   END FOR
11-   IF similarElements are smaller than THRESHOLD2
    THEN
12-     MOTIONLEVEL = 2;  TERMINATE HERE
13-   ELSE IF similarElements are smaller than
    THRESHOLD1 THEN
14-     MOTIONLEVEL = 1; END IF
15- END FOR
```

Figure 24: GOP analysis Algorithm

5.2 METHODOLOGY

5.2.1 Encoder

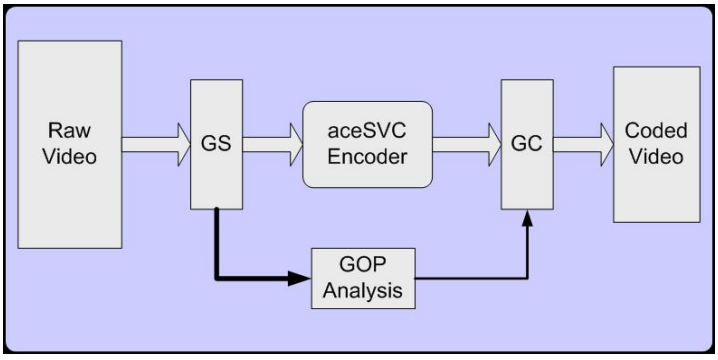


Figure 25: SCC Encoder

A block diagram of the SCC encoder is shown in Figure 25. The GOP Selector (GS) block picks up the raw data of one GOP and forwards it to the encoder and the GOP Analysis (GA) block for processing. The encoder generates a compressed GOP with all the scalability layers as predefined in the scalability tree through an encoding parameter file. The GA block processes the GOP data and generates analysis metadata as described in section 5.1.3. This metadata is passed to the GOP Collector (GC) which associates it with the encoded GOP. This process is repeated until each GOP of raw video is encoded and put into the encoded bit-stream.

5.2.2 Extractor

The function of an SVC extractor is to extract an encoded video bit-stream with the required scalability level. In new architecture of extractor, again the GS and GC block are used with Extraction Control (EC) block as shown in Figure 26. The GS block will pick up compressed GOP data from the main encoded video bit-stream and forward it to the SVC extractor and the Extraction Control (EC) block. Extraction Control (EC) block extracts the GOP analysis metadata 'μ' produced by the GOP Analysis (GA) block on the encoder side. This information is passed to the SVC extractor for the extraction of appropriate scalability layer. The extracted GOP is then passed to the GC block which manages the extracted GOP for the final extracted bit-stream. This process is repeated for all the GOPs.

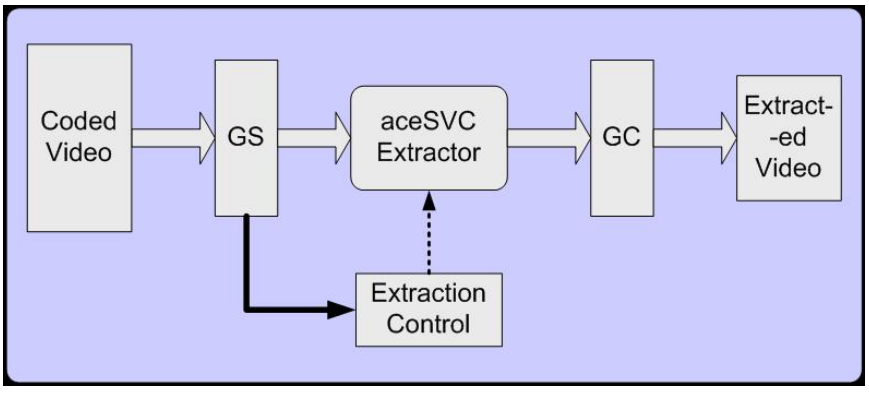


Figure 26: SCC Extractor

5.2.3 Decoder

As in the case of the architecture of the encoder and extractor, the decoder architecture is also changed. Again, the GS and GC blocks are deployed, which perform the same functionality as mentioned for encoding and extraction. The decoder recovers the video in an uncompressed form. Decoded video contains GOPs with different scalability levels according to the event present in the GOP. After decoding the compressed bit-stream, post-processing is performed to enable the smooth viewing of the decoded video.

5.3 Model of the System

The above discussion can be explained through simple mathematical equations. So, the mathematical model of the modified scalable video coding system is presented in the following subsections.

5.3.1 Encoder

The model of SCC encoder is shown in Figure 27. The raw video data can be represented with equation (5.1). This is the input to this stage. The encoding (Enc) and GOP analysis (A) operations can be represented with equations (5.3) and (5.4), respectively.

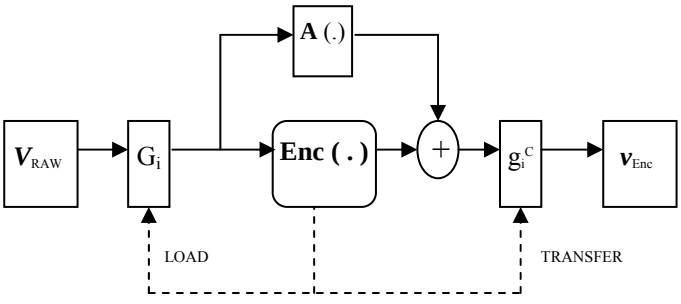


Figure 27: Model of SCC Encoder

$$g_i^{Enc} = \text{Enc}\,(G_i\,) \qquad \textbf{(5.3)}$$

where,

$$G_i \in V_{RAW}$$

$$\mu = A(G_i) \quad \textbf{(5.4)}$$

The compressed GOP (g_i^{Enc}) with added GOP analysis information (μ_i) is represented with equation (5.5).

$$g_i^c = \mu_i + g_i^{Enc} \quad \textbf{(5.5)}$$

where,

$$g_i^c \in v_C$$

such that:

$$v_C = \{ g_1^c, g_2^c, g_3^c, \ldots, g_N^c \} \quad \textbf{(5.6)}$$

Equations (5.3) through (5.6) describe the complete process of encoding for the modified system.

5.3.2 Extractor

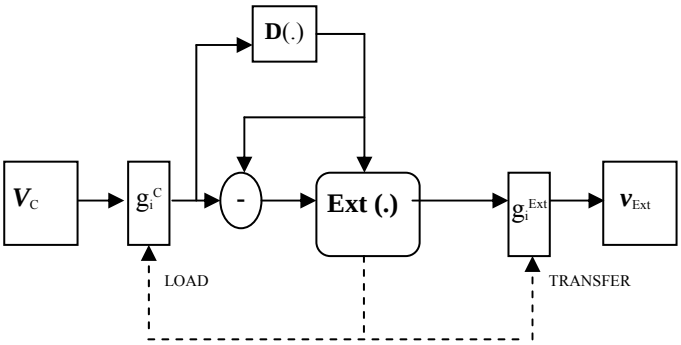


Figure 28: Extractor Model

The model of Extractor is shown in Figure 28. The input to this stage is equation (5.6). Each GOP of equation (5.5) is forwarded to the decision extraction (D) and the GOP extraction block according to equations (5.7) and (5.8), respectively.

$$\mu_i = D(g_i^c) \quad \textbf{(5.7)}$$

$$g_i^{Enc} = g_i^c - \mu_i \quad \textbf{(5.8)}$$

After executing equations (5.7) and (5.8), the SVC extractor executes equation (5.9) given below and the extracted GOP is put into the final extracted bit-stream given in equation (5.10).

$$g_i^{Ext} = \text{Ext}\,(g_i^{Enc}) \quad \textbf{(5.9)}$$

where,

$$g_i^{Ext} \in v_{Ext}$$

such that:

$$v_{Ext} = \{\, g_1^{Ext}, g_2^{Ext}, g_3^{Ext}, \ldots, g_N^{Ext} \} \quad \textbf{(5.10)}$$

5.3.3 Decoder

Decoding is final step of the codec. Its input is an extracted video given in equation (5.10). The model can be easily described with equations (5.11) and (5.12). Here, the G_i^{Dec} is a decoded GOP from the compressed GOP.

$$G_i^{Dec} = \text{Dec}\,(g_i^{Ext}) \quad \textbf{(5.11)}$$

$$V_{DEC} = \{\, G_1^{Dec}, G_2^{Dec}, G_3^{Dec}, \ldots, G_N^{Dec} \} \quad \textbf{(5.12)}$$

The decoded video V_{DEC} is visually similar to V_{RAW} , given in equation (5.1). This can be described as:

$$V_{DEC} \equiv V_{RAW} \quad \textbf{(5.13)}$$

5.4 Functionality Evaluation

As aforementioned, extraction of GOP with different scalability features depends on the level of motion present in the sequence. The modified system has full support for

the selection of a different scalability layer for multiple motion levels. The scalability tree for encoding can be defined as per requirement. For this experimentation, the tree shown in Figure 29 has been used.

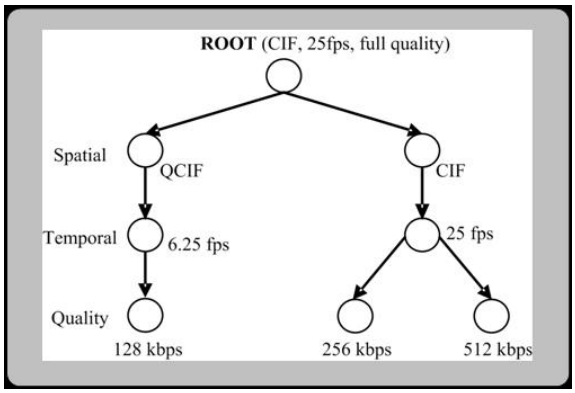


Figure 29: Scalability Tree

Root node contains full scalability levels i.e. the raw video. For the encoding of the video, two spatial and two temporal levels are defined while there are three quality levels. So, three types of bit-streams, (QCIF, 6.25fps, 128kbps), (CIF, 25fps, 256kbps) and (CIF, 25fps, 512kbps), can be extracted.

Figure 30: No Motion: static scene

The experiment is carried out on a typical surveillance video. The video contains all types of motion scenarios. When there is no motion in the video, it is set to motion

level 0, while for slight movements it is set to motion level 1 and for all other higher movements, motion level 2 is used.

Figure 31: Motion Level 1: slight movement

The original uncompressed video has CIF resolution and 25 fps. The portion of the video which has no motion (motion level 0) is extracted at QCIF spatial resolution, 6.25 fps temporal resolution and 128 kbps quality resolution, as shown in Figure 30. The portion of video which has motion level 1 is extracted at CIF spatial, 25 fps temporal and 256 kbps quality resolution, as shown in Figure 31.

Figure 32: Motion Level 2: large movement

Finally, the portion of video which has motion level 2 has been extracted at CIF spatial, 25 fps temporal and 512 kbps quality resolution as shown in Figure 32.

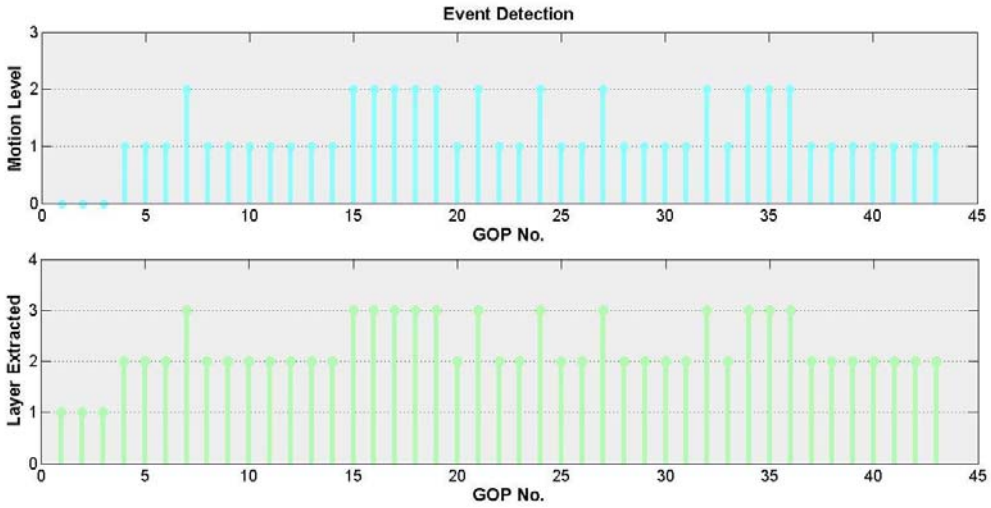


Figure 33: GOP Motion Analysis

The graph in Figure 33 shows motion levels present in the surveillance video used for the experiment. Level 0 is for no movement, level 1 for slight movements and level 2 for greater movements. The second graph in Figure 33 shows final extracted video with scalability levels for each GOP. The second graph testifies the analysis presented in the first graph in Figure 33. In the second graph of Figure 33, layer 1 is for QCIF, 6.25 fps and 128 kbps; layer 2 is for CIF, 25 fps and 256 kbps; while layer 3 is representing CIF, 25 fps and 512 kbps.

5.5 Surveillance Centric Coding

The architecture of a generic SCC system is outlined in Figure 34 [103].The work presented in this section focuses on the use of video content analysis (VCA) to drive the encoding process. By this approach the use of available resources is optimised according to the requirements of surveillance applications. In the proposed approach, the SCC encoder communicates with the VCA modules and performs encoding by rate-optimisation according to events as specified by the VCA. The VCA can also be used on the decoder-side for off-line processing, e.g., car plate recognition, face detection, etc. Therefore, the question behind this work is how to exploit the information resulting from the VCA to tailor the coding and transmission or

streaming of the video signal. Clearly, this cannot be achieved with conventional coding technology without complex transcoders. It can however be achieved if fine granularity scalable coding technology is applied.

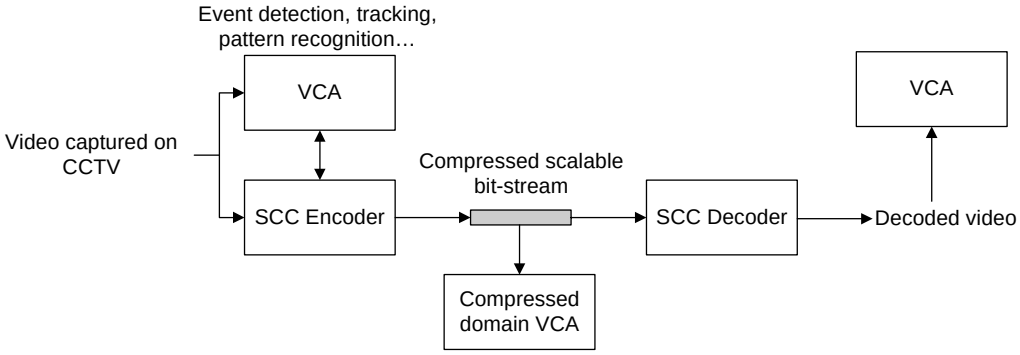


Figure 34: A generic SCC system.

5.5.1 Event-Based Encoding of Surveillance Video

As mentioned in the introductory section, the basic principle behind the presented work is to use different encoding settings for segments of the surveillance video that show different levels of activity. For this purpose we classify the surveillance video into temporal segments that contain essentially static scenes and segments that show some level of motion activity. To perform this classification, background subtraction and the tracking module from the work of Stauffer and Grimson [70] is used as VCA. Output of this module dictates the quality / spatio-temporal resolution of the encoded content. As the VCA module plays a vital role to the final outcome of the SCC system, therefore it is explained briefly in the following subsection.

5.5.2 Video Content Analysis

An adaptive background subtraction method based on mixture of Gaussians [70] is used. This method is able to deal robustly with lightning changes, bimodal background like swaying trees and introduction or removal of objects from the scene. The value of each pixel is matched against weighted Gaussians of the mixture. If the pixel value is within 2.5 standard deviations of any Gaussian distribution then the mean value and standard deviation of the corresponding Gaussian are updated. If the

pixel value is not within 2.5 standard deviations of any distribution then the least probable distribution is replaced by the new distribution. The mean value of the new distribution is set as the value of the current pixel and its initial variance is set to a high value. Weights are continuously updated for each distribution of the mixture.

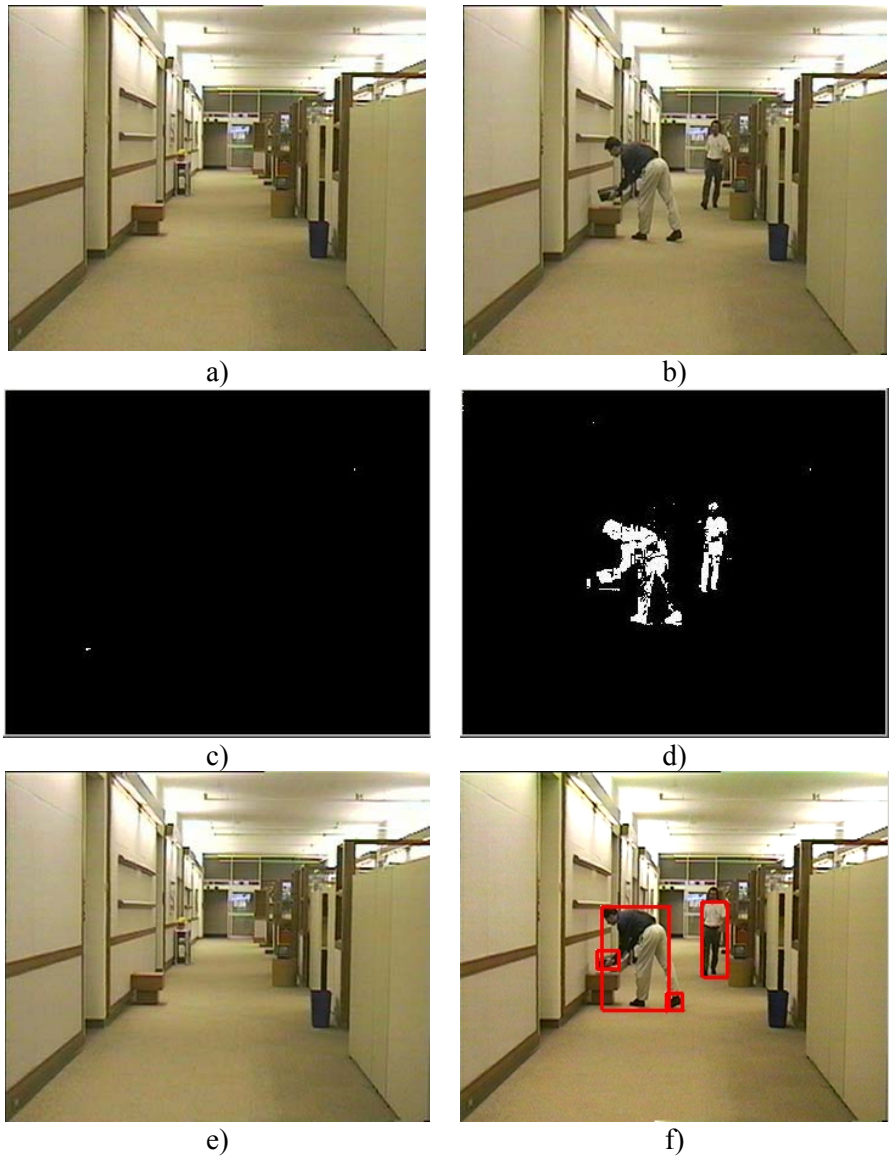


Figure 35: Background subtraction. a) 3-rd and b) 110-th frame of the hall sequence. Result of the background subtraction for the c) 3-rd and d) 110-th frame. Result of tracking for the e) 3-rd and f) 110-th frame

At each time instance Gaussians of the mixture that represent the background are identified according to the predefined threshold. Pixels whose value is not within 2.5 standard deviations of the Gaussians representing the background are declared as the foreground. Foreground pixels can then be segmented into regions and tracked throughout the sequence. The output of the background subtraction and tracking module is illustrated in Figure 35.

5.5.3 Workflow of the System

The proposed system for application in SCC coding is outlined in Figure 36. At each time instance the encoder communicates with the VCA module. When the input video is essentially static the output of the background subtraction does not contain foreground regions. This can be used to signal to the encoder to encode the captured video at a low spatio-temporal resolution and quality. Encoding parameters in this case can be defined by the user or they can be chosen automatically by the system. This allows, for instance, encoding and/or transmitting of the portions of the video containing long, boring, static scenes using low quality frame-rate and spatial resolution. On the other hand, when some level of activity in the captured video is detected, the VCA module notifies the encoder to automatically switch encoding to a desired much higher spatio-temporal resolution and quality video. Therefore, decoding and use of the video at different spatio-temporal resolutions and qualities corresponding to different events is achieved from a single bit-stream, without multicasting or complex transcoding. Moreover, additional optional adaptation to a lower bit-rate is also possible without decoding the video. This is, for instance, very useful in cases where the video has to be delivered to a device with a low display capability. Using this approach, the bit-rate of parts of the video that are of low interest is kept low while the bit-rate of important parts is kept high. In many realistic applications it can be expected that large portions of the captured video have no events of interest. Thus, the proposed model leads to significant reduction of resources without jeopardizing the quality of any off-line event detection module.

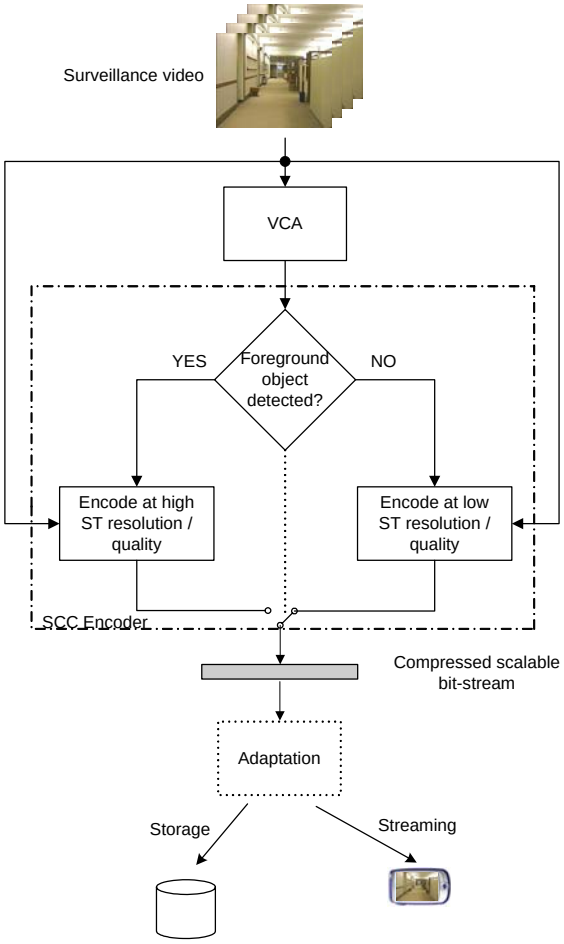


Figure 36: The workflow of the event-based encoding.

As an added side effect, critical metadata encoding events detected by the VCA can be interleaved with the bit-stream. Therefore events of interests can easily be identified in the compressed domain just by reading the corresponding metadata extracted directly from the video stream.

5.6 Performance Results

Performance of the proposed SCC framework has been evaluated using different typical surveillance sequences listed in Table 2. Encoding was performed on the GOP by GOP basis, i.e. the smallest group of frames which can be encoded / adapted according to the output of VCA is defined by the GOP size. In our experiments, we use three different GOP sizes: 8, 16 and 32.

Seq.	Frame Size	Frame Rate	Frames
Bridge	352 × 288	30	2100
Dance	352 × 288	30	500
Hall	352 × 288	30	300
LightOffOn	352× 288	30	500
LightOnOff	352× 288	30	500
Parking	352 × 288	30	2100
Road Car	352× 288	30	500
Street	352 × 288	30	750

Table 2: Surveillance sequences

The proposed framework is evaluated with respect to its flexibility and efficiency. Criteria given in Table 3 are used to adapt the encoding of surveillance sequences with respect to different spatio-temporal resolutions and qualities. If the VCA does not detect an event in the particular portion of the sequence, it passes the information to the SCC encoder that performs the encoding process either at low spatial resolution (width and height of the frame is reduced to half i.e. spatial adaptation – QCIF), frame-rate (temporal adaptation – half of original frame-rate) or the combination of all three scalability directions (spatial, temporal, quality), as given in Table 3. When the event is detected by the VCA, encoding is switched to original resolution and frame-rate and high quality.

Event	Adaptation	Bit-rate
Essentially static scene	Spatial	144 kbps
	Temporal	288 kbps
	Quality	128 kbps
	Combined	100 kbps
Event detected (Man/truck/boat)	Full spatio-temporal resolution	512 kbps

Table 3: Adaptation bit-rates

Table 4, Table 5 and Table 6 show byte savings for each tested sequence, using the SCC encoder with different types of rate adaptation. The byte savings are shown

relative to the corresponding sequence encoded at full resolution, frame-rate and high quality for the whole length. The relative byte saving is calculated as:

$$rbbs = (1 - NB_{SCC} / NB_{Conv}) \times 100\ \% \qquad \textbf{(5.14)}$$

where 'NB_{SCC}' represents the number of bytes of the compressed sequence using the SCC approach from Figure 36 and 'NB_{Conv}' represents number of bytes of the compressed sequence encoded at full resolution, frame-rate and quality for the whole length.

Seq.	Adaptation for %age Byte Savings			
	Spatial	**Temporal**	**Combined**	**Quality**
Bridge	65.58	39.91	73.43	68.42
Dance	28.74	17.50	32.16	30.01
Hall	3.83	2.33	4.30	4.00
LightOffOn	42.64	25.93	47.73	44.49
LightOnOff	10.37	6.31	11.61	10.82
Parking	52.29	31.83	58.54	54.59
Road Car	71.90	43.75	80.46	75.00
Street	4.60	2.80	5.15	4.80

Table 4: GOP size 8: Relative byte savings

The relative byte saving '*rbbs*' using different GOP sizes is shown in Table 4, Table 5 and Table 6 for GOP 8, 16 and 32, respectively. From these tables, it can be observed that the compression gains for the "Hall" sequence are rather small. This is because throughout the whole sequence some level of activity is present and therefore almost the whole sequence is encoded at the original spatio-temporal resolution and high quality. This is clearly evident in Table 6 where '*rbbs*' for the 'Hall' sequence is zero which indicates all the frames are encoded at full spatio-temporal resolution. So, the GOP size affects the efficiency of the SCC. For other surveillance sequences, significant bit-rate savings can be observed, especially in "Bridge" and "Parking"

sequences because of motion activity present in a short interval of time for these sequences.

Seq.	Adaptation for %age Byte Savings			
	Spatial	Temporal	Combined	Quality
Bridge	62.56	38.08	70.05	65.28
Dance	27.60	16.80	30.90	28.80
Hall	3.83	2.33	4.29	4.00
LightOffOn	41.48	25.25	46.44	43.28
LightOnOff	9.22	5.61	10.32	9.62
Parking	52.02	31.66	58.24	54.29
Road Car	71.88	43.75	80.46	75.00
Street	3.07	1.87	3.43	3.20

Table 5: GOP size 16: Relative byte savings

Seq.	Adaptation for %age Byte Savings			
	Spatial	Temporal	Combined	Quality
Bridge	58.73	35.75	65.75	61.28
Dance	23.00	14.00	25.75	24.00
Hall	0.00	0.00	0.00	0.00
LightOffOn	41.48	25.25	46.44	43.28
LightOnOff	9.22	5.61	10.32	9.62
Parking	51.47	31.33	57.63	78.52
Road Car	71.87	43.75	80.47	75.00
Street	0.00	0.00	0.00	0.00

Table 6: GOP size 32: Relative byte savings

Decoded visual images of the proposed SCC for the “Hall” sequence are presented in Figure 37 and for the “Parking” sequence in Figure 38. The first row shows original frames. The second row represents the binary mask of the original video, which is the output of the background subtraction module. The third row shows the reconstructed sequence whose essentially static segments were encoded at lower spatial resolution.

Fourth row represents the temporally adapted sequence. Note that only one of the two consecutive original frames is kept in the adapted portion of the sequence. The last row shows the combined scalability, i.e. reduction of spatio-temporal resolution and quality.

Figure 37: Frames of the reconstructed "Hall" sequence obtained. First row: the original frames. Second row: binary mask. Third row: Spatial adaptation. Fourth row: Temporal adaptation. Fifth row: Combined adaptation.

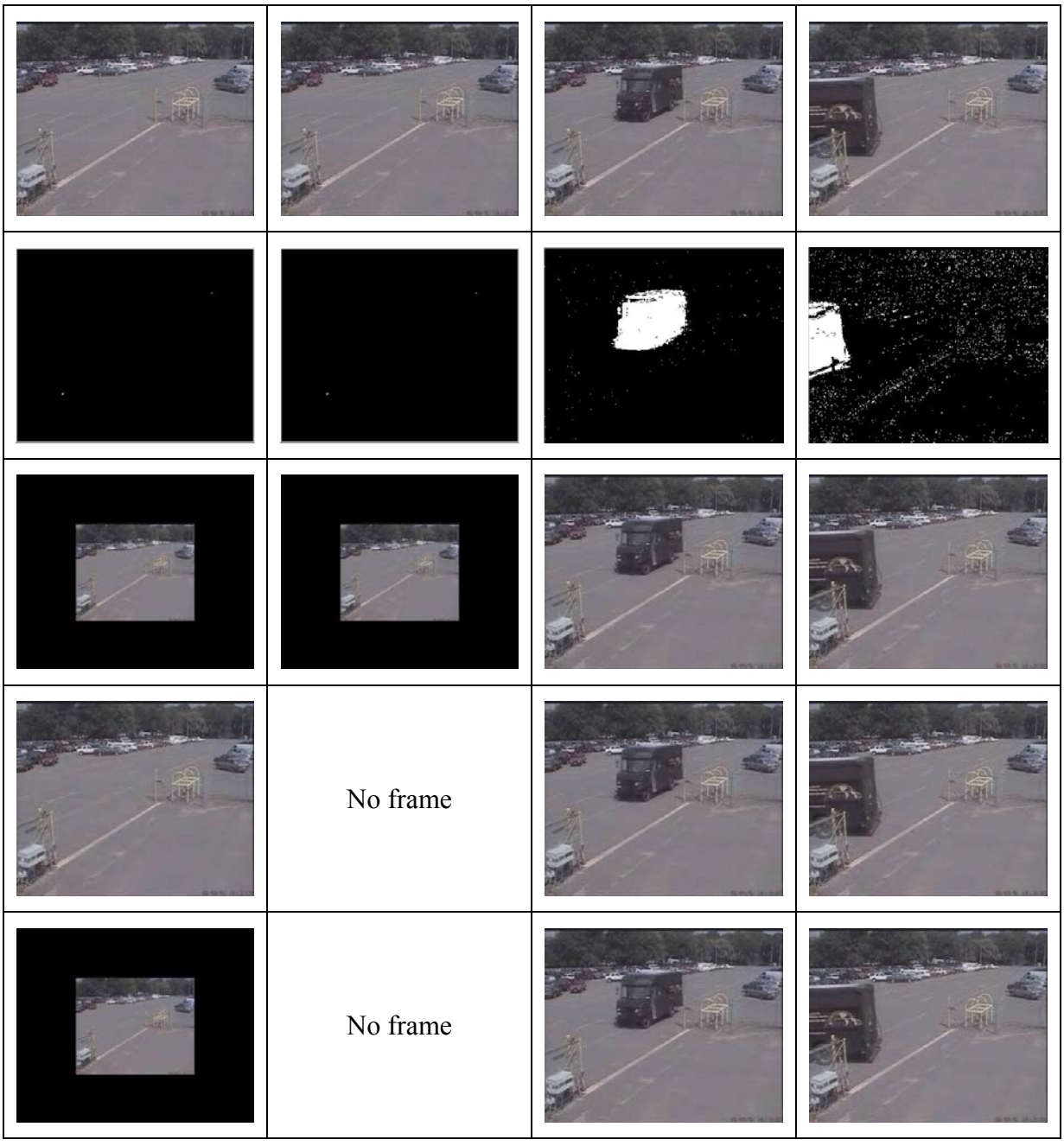


Figure 38: Frames of the reconstructed "Parking" sequence obtained. First row: the original frames. Second row: binary mask. Third row: Spatial adaptation. Fourth row: Temporal adaptation. Fifth row: Combined adaptation.

5.7. Foreground Based SCC

After event based surveillance centric coding, a novel approach for foreground based surveillance centric coding (FBSCC) is proposed. Several attempts have been made in the past to compress the surveillance videos retaining only the information relevant to surveillance applications. In the work of Vetro et al [16], surveillance videos have been encoded using object-based MPEG-4 after subtracting the background. The single background image is compressed using frame-based technique. After decoding the compressed sequence and the background image, the background image is repeated for each frame. In the work of Nishi and Fujiyoshi [101], pixel state analysis is performed to detect background and foreground objects. Further analysis distinguishes the foreground object pixel as transient or stationary pixels. The transient pixels of foreground objects are compressed using Lempel-Ziv-Huffman (LZH) codec. For stationary pixels, the colour intensity is restored by referring to the same pixel location in the last frame. In the work of Hakeem et al [102], an object-based video coding framework for video sequences obtained from the static camera has been presented. The developed system detects and tracks objects in the scene and learns the appearance model for each object.

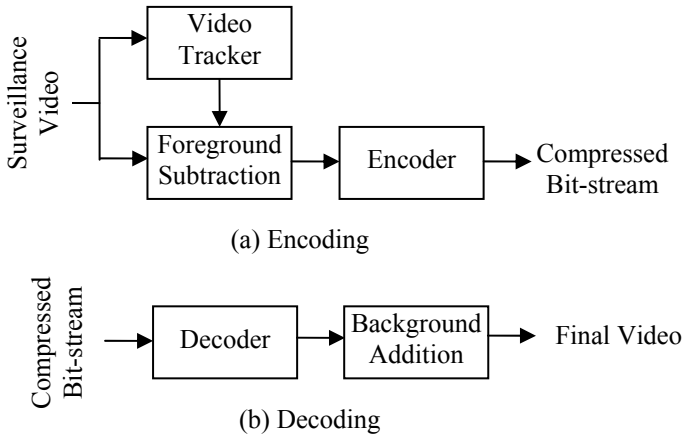


Figure 39: Architecture of implemented system

In this work, an alternative approach is proposed to reduce the bit-rate of the encoded video segments that are irrelevant from the surveillance standpoint. The proposed approach combines video tracking, foreground subtraction and scalable video coding. A novel approach for foreground extraction is proposed where the information

provided by the surveillance video tracker is used. This ensures that only the required information is extracted for compression. This approach is independent of shape coding techniques required in the case of MPEG-4 object-based coding. Also, this approach has the benefit of using a fully scalable video codec which can be used to fulfil further requirements of storage and transmission.

5.8. Methodology

The architecture of the implemented system is shown in Figure 39. The surveillance video captured from the CCTV camera is presented to a surveillance video tracker and foreground subtraction module of the system. The foreground objects are tracked by forming rectangular boxes. The output of the video tracking module for the 'Hall' video is illustrated in Figure 40.

Figure 40: Object tracking in frame 122 of hall video

The information about each box dimensions in each frame of the video is passed to the foreground extraction module of the proposed system. Based on the dimensions of the tracking boxes, pixels bounded by these boxes are retained as foreground and the rest of the frame is given a black colour. The algorithm for the foreground subtraction module is presented in Table 7.

The algorithm is designed for YUV video format with 4:2:0 chroma sub-sampling. To accommodate U and V components of the video, the dimensions of the tracking

boxes are scaled down. The algorithm handles multiple tracking boxes for each frame of the surveillance video.

```
for (each frame of input video)
   for hy=0 to FrameHeight – 1
      for wy=0 to FrameWidth – 1
         o  y_in [hy][wy] = Y component of the frame
         o  y_out [hy][wy]= 0
      end for
   end for
   for hu=0 to FrameHeight/2 – 1
      for wu=0 to FrameWidth/2 – 1
         o  u_in [hu][wu] = U component of the frame
         o  u_out [hu][wu]= 128
      end for
   end for
   for hv=0 to FrameHeight/2 – 1
      for wv=0 to FrameWidth/2 – 1
         o  v_in [hv][wv] = V component of the frame
         o  v_out [hv][wv]= 128
      end for
   end for

   for (each box in the frame)
      Get (Dimensions of the rectangular box)
      for hy=0 to FrameHeight – 1
         for wy=0 to FrameWidth – 1
            if (hy and wy are within box)
               o  y_out [hy][wy]= y_in [hy][wy]
         end for
       end for
      for hu=0 to FrameHeight/2 – 1
         for wu=0 to FrameWidth/2 – 1
            if (hu & wu are within scaled dimensions of box)
               o  u_out [hu][wu]= u_in [hu][wu]
         end for
       end for
      for hv=0 to FrameHeight/2 – 1
         for wv=0 to FrameWidth/2 – 1
            if (hv & wv are within scaled dimensions of box)
               o  v_out [hv][wv]= v_in [hv][wv]
         end for
       end for
   end for

   Save (y_out, u_out and v_out as a YUV frame)
end for
```

Table 7: Algorithm for foreground subtraction

Figure 41 presents the tracked frame in Figure 40 after the application of foreground subtraction algorithm. The very first frame of the video is trained as the background while the subsequent frames contain only the foreground pixels identified through the bounding boxes. By using the bounding boxes, only the information of interest from the surveillance standpoint is obtained and presented for encoding.

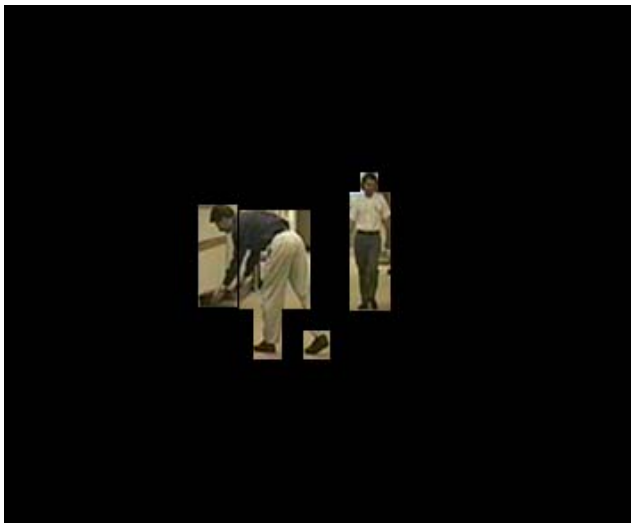


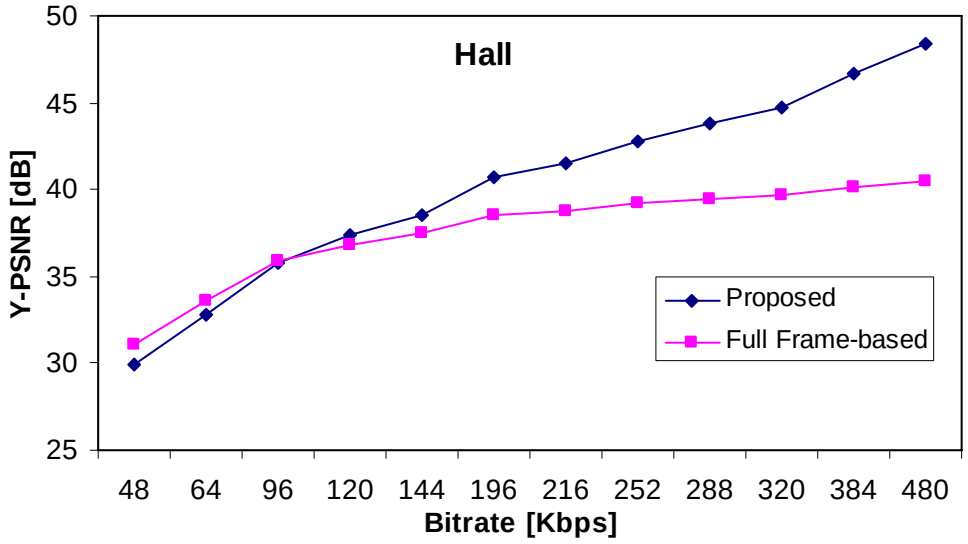
Figure 41: Foreground subtracted frame 122 of hall video

For actual encoding of the subtracted video, a wavelet-based scalable video codec – aceSVC [71] is employed. The architecture of the aceSVC features spatial, temporal, quality and combined scalability. After decoding the compressed bit-stream of foreground subtracted sequence, first frame is added as the background of each frame to generate a surveillance sequence for better human visual understanding.

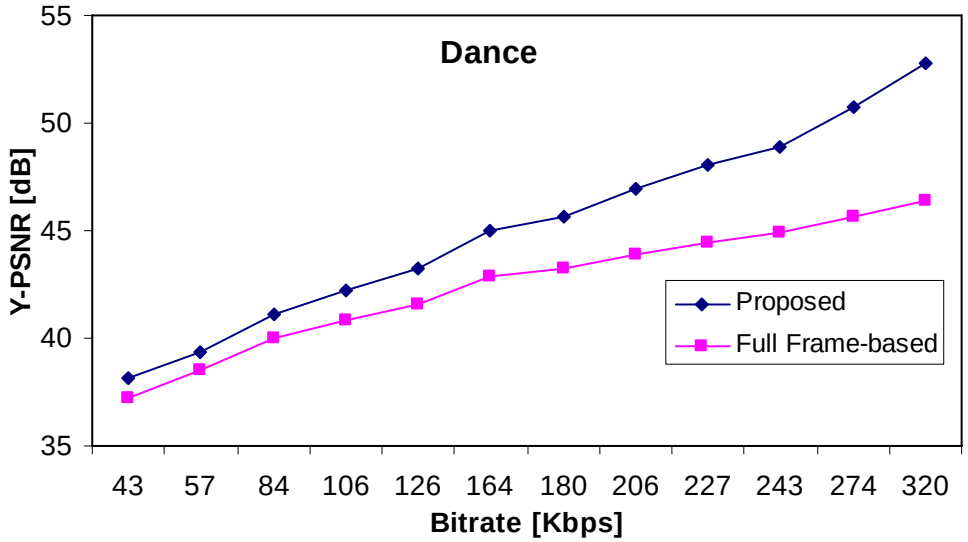
5.9. Experimental Results for FBSCC

Performance of the proposed framework has been evaluated using four different typical surveillance sequences: the “Bridge” with 2100 frames, “Hall” with 300 frames, “Dance” with 500 frames and “Street” with 750 frames. All the sequences have a resolution of 352×288 pixels (CIF) and frame-rate of 30 Hz. All of these sequences have a static background throughout the length of the sequences. The ‘Bridge’ sequence has a small boat appearing for a short time. So, out of 2100 frames, small numbers of frames have foreground object. In the ‘Dance’ sequence, an

animated person is dancing with fast arms and legs movement. In the 'Street' sequence, with an outdoor street background, different animated objects are moved through the street. In the 'Hall' sequence, two people are walking in opposite directions in a corridor.

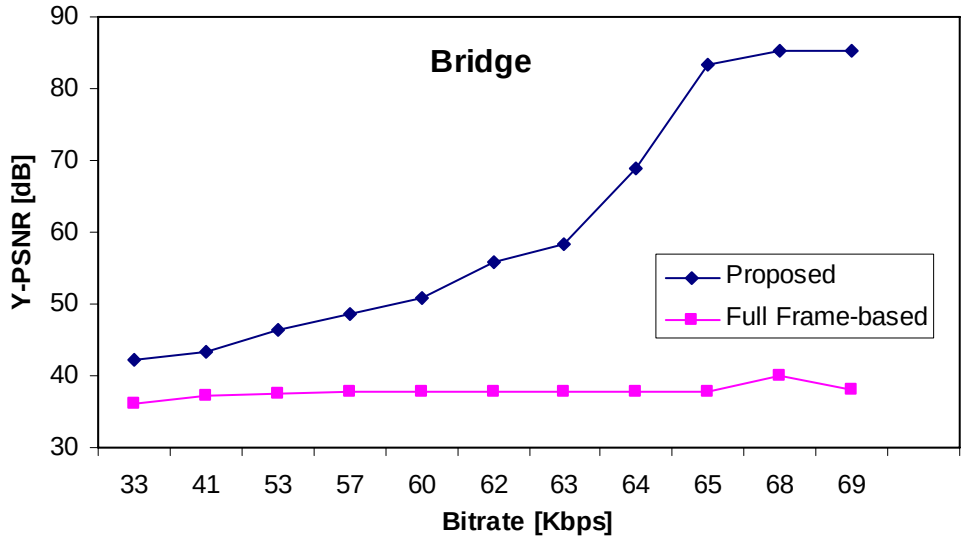


(a)

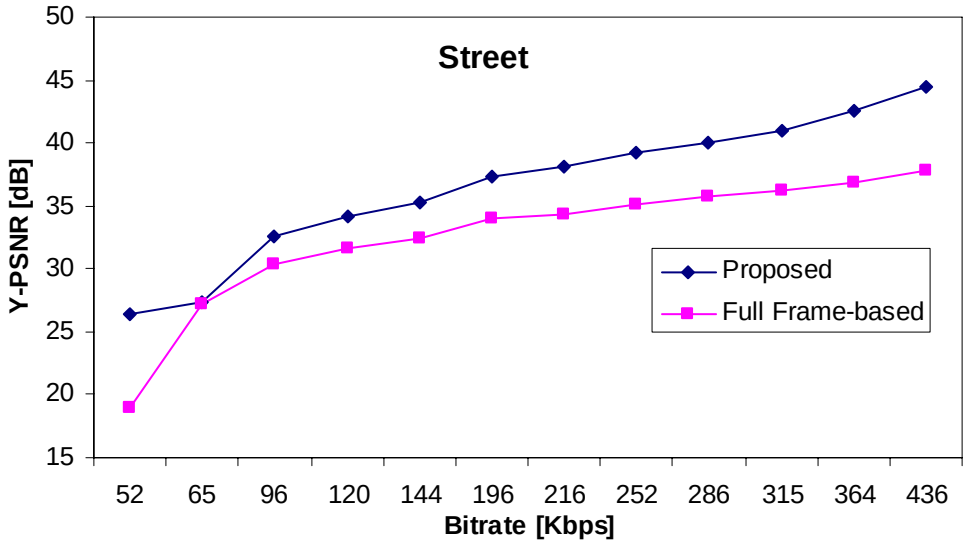


(b)

Figure 42: PSNR Results for Indoor surveillance sequences



(a)



(b)

Figure 43: PSNR results for outdoor surveillance sequence

In the Peak Signal to Noise Ratio (PSNR), described in section 2.2.1, results shown in Figure 42 and Figure 43, the legend Full Frame-based represents the full frame-based surveillance video without any pre-/post- processing, compressed using aceSVC. For the PSNR evaluation of the proposed strategy, the original surveillance sequence is tracked and foreground subtracted using box information. After foreground subtraction, again the background frame is added. This generates a raw sequence

subjectively similar to the decoded and background added sequence. This new raw sequence is used to calculate the PNSR values of the proposed strategy.

(a)

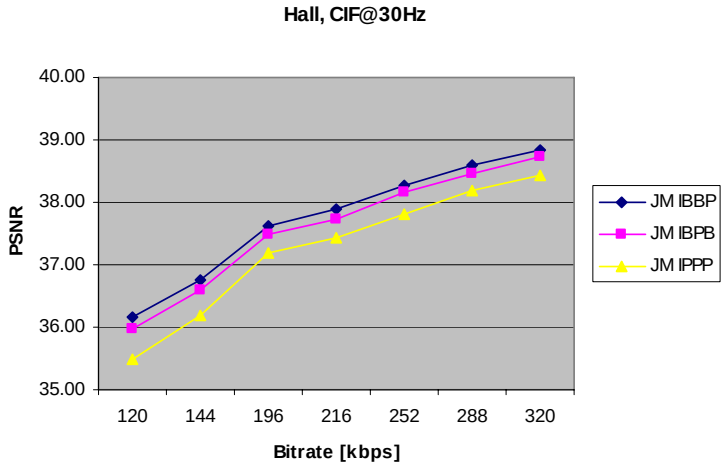
(b)

Figure 44: Subjective comparison of reconstructed frame 122 of Hall sequence at same bit-rate (a) Full Frame-based (b) Foreground-based

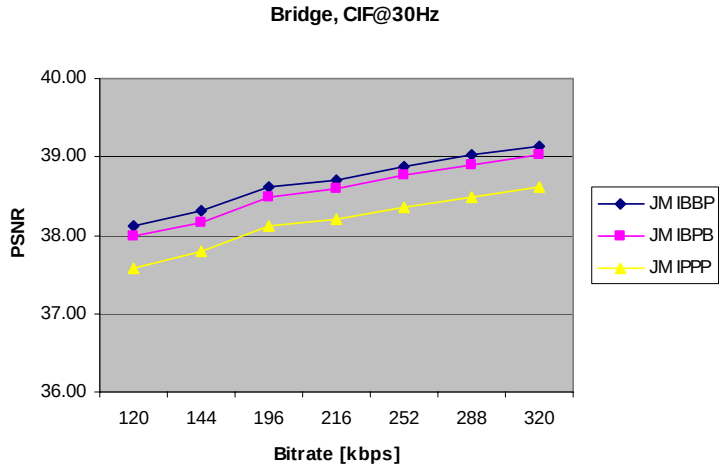
Results show that the proposed strategy has a much better Rate-Distortion performance. The subjective comparison of the decoded frames is given in Figure 44. This proves that all the information important from the surveillance standpoint is maintained.

5.10 Related Test Results

The overall aim of this research work is to develop different algorithms and techniques for the Rate-Distortion optimized coding of the surveillance videos. First, the state-of-the-art coding standards are applied on the surveillance videos to evaluate their performance. The H.264 reference software, JM 12.4 version, is used in its different configurations and applied to the 'Hall' and the 'Bridge' videos. The result of this experiment is shown in Figure 45.



a) Hall video

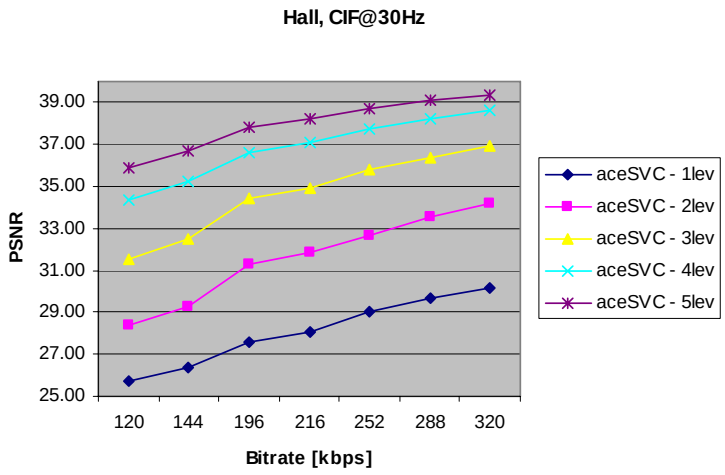


b) Bridge video

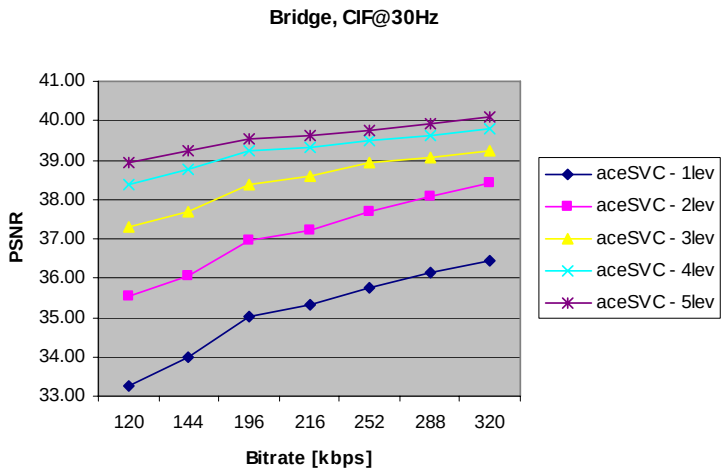
Figure 45: H.264 R-D plot

The JM configuration of 'IBBP' has the best PSNR results for both Hall and Bridge videos. But this configuration has higher complexity because of the higher number of 'B' frames. Results for Bridge video are slightly better than the Hall video. The Bridge video has most of the frames with very minor environmental changes in the scene while the Hall video has a lot of motion activity.

Next step, a wavelet based video codec, aceSVC, has been applied on the Hall and Bridge videos. In aceSVC, the temporal redundancy present in the video is removed using the Motion Compensated Temporal Filtering (MCTF).



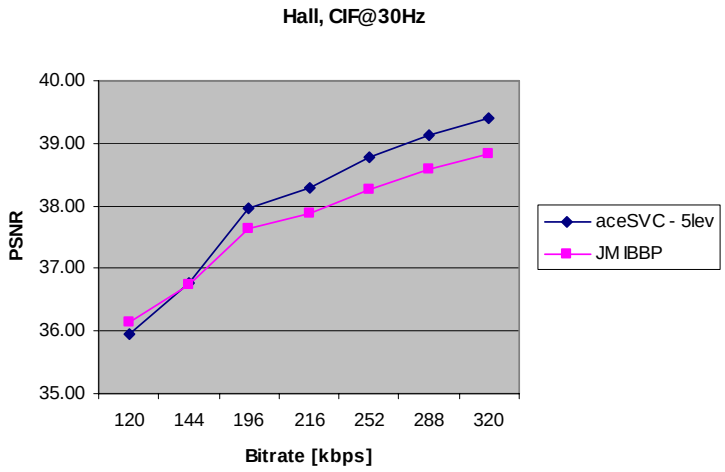
a) Hall video



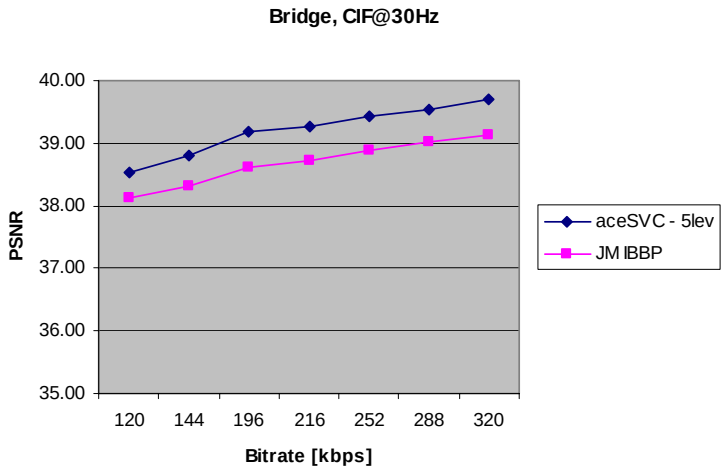
b) Bridge video

Figure 46: aceSVC R-D plot

In this test, different levels of temporal filtering are set to evaluate the R-D performance of the aceSVC. The results of this test for the Hall and the Bridge videos are shown in Figure 46. On the basis of these results, it is evident that for surveillance videos, the higher temporal level filtering has a better R-D performance. As in H.264, the Bridge video has better results for aceSVC as well. The Bridge video has very high temporal redundancy which MCTF has utilized to improve the R-D performance. After evaluating the block-based H.264 and wavelet-based aceSVC for their different configurations, the comparison between results of the best configuration of the each codec is done.



a) Hall



b) Bridge

Figure 47: aceSVC vs JM

The R-D performance comparison between aceSVC and JM is shown in Figure 47 for the Hall and the Bridge videos, respectively. The result for the Hall video shows that JM performed better than aceSVC at low bit-rates. After the 144 kbps and higher bit-rates, aceSVC has produced better results than JM. While for the Bridge video, the R-D performance of aceSVC is better than JM for all bit-rates.

Motion estimation techniques in any codec, block-based or wavelet based, have the highest complexity as compared to all other modules of the codec. They consume a lot of processing power causing a high latency. For most of the surveillance scenarios, there is very little motion activity in the videos. Furthermore, the static background throughout the video reduces the effect of this motion activity.

5.11. Conclusions

In this chapter, the coding system that performs rate optimisation and adaptation in surveillance applications was proposed. This was achieved by an interaction between a Video Content Analysis (VCA) module and the Wavelet-based Scalable Video Coding. A modified architecture for event based SVC was proposed which was efficiently adapted to the surveillance application. GOPs were created in such a way that there was no interlink between the GOPs. The GOP analysis approach was used to detect the motion or event effectively. This analysis was used to switch the video to different scalability levels according to significance of the event in the GOP. Experimental results showed that the modified architecture successfully controlled the scalability features of GOPs after detecting motion events. Time segments containing events relevant to surveillance applications are detected by the VCA and encoded using a higher spatio-temporal resolution and a better quality. Other portions of the video are encoded at low spatio-temporal resolution and / or quality. Experimental results show that significant bit-rate reductions can be achieved by using the proposed approach.

In another approach, a novel technique to obtain the foreground objects relevant to surveillance applications using the information of a surveillance video tracker was presented. A fully scalable video codec has been used to compress the foreground-only sequence. A very high compression rate is achieved by using the proposed technique. Performance of the implemented system is compared with normal full

frame based coding. Subjective quality and PNSR results obtained through experimental evaluation show that significant bit-rate reduction can be achieved by using the proposed approach. Use of fully scalable video codec provides the flexibility to achieve further reduction of bit-rate while still preserving the required surveillance information.

Chapter 6

Selective Motion Estimation

6.1 Introduction

In surveillance applications, video captured by the CCTV is usually encoded using conventional techniques, such as H.264/AVC. These techniques have been developed in view of conventional videos. With a growing number of surveillance system deployments, there is a need to introduce surveillance centric coding techniques. The goal of this work is to propose an efficient motion estimation approach specific to surveillance videos. During the encoding process of videos, motion estimation is performed to find the motion vectors for the best matched block in a search window. While decoding video, these motion vectors are used to reconstruct the original blocks with their best matched block in an already decoded frame. Encoding complexity is dominated by the ME if full search (FS) is used as the BMA. FS matches all possible displaced candidate blocks within the search window to find a block with the minimum Block Distortion Measure (BDM).

Several fast BMAs have been introduced to beat FS in terms of computational complexity. These include the new three step search (N3SS) [59], four-step search (4SS) [60], diamond search (DS) [61], kite-cross diamond search (KCDS) [66], and modified DS (MODS) [67], etc. In this chapter, a novel approach to reduce computational complexity for encoding surveillance videos has been proposed. The proposed approach utilizes a real-time background subtractor (BGS) [70] to detect the presence of the motion activity in the sequence. In typical surveillance videos, a scene remains static for a long period of time. Performing motion vector search for these frames is wastage of computing resources. A motion vector (MV) search is performed only for frames which have some motion activity identified by the BGS.

6.2 Selective Motion Estimation

The generic architecture of the implemented system is shown in Figure 48. A surveillance video is fed to background subtraction and video encoding modules of the system. The real-time background subtractor, motion detection block in Figure 48, detects motion activity present in the sequence. This information is passed onto the motion estimation module of the encoder. The motion estimation module utilizes the motion detection information to perform selective motion estimation. After motion compensated temporal filtering (MCTF) step, spatial transformation is performed to remove the spatial redundancies. Finally, entropy coding techniques like CABAC are used to improve compression efficiency.

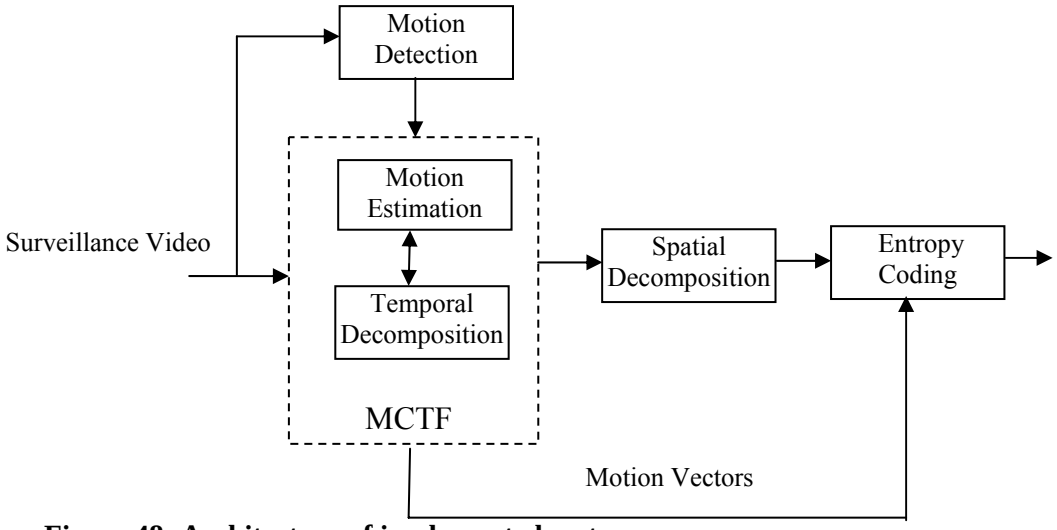


Figure 48: Architecture of implemented system

6.2.1 Real-Time Background Subtraction

A motion detection module must be efficient in terms of utilizing processing power otherwise; the complexity of the motion estimation module shall be reduced at the cost of increased complexity for motion detection. A real-time video background subtraction module based on the Gaussian mixture model [70] is used to detect motion activity present in the video. This method is able to deal robustly with light changes, bimodal background like swaying trees and introduction or removal of objects from the scene. The value of each pixel is matched against weighted Gaussians of mixture. Those pixels are declared as foreground whose value is not within 2.5 standard deviation of the background Gaussians. Foreground pixels are grouped into segmentation regions and bounded by rectangular boxes throughout the length of the sequence. The output of the BGS module for hall video is illustrated in Figure 49. Pixels which are static for a number of frames are modelled as background; therefore they do not fall within the boundary of boxes. Presence of the bounding box is an indication of motion activity present in the frame. This indication is used to perform selective motion estimation.

Figure 49: BGS result for frame 122 of hall video

6.2.2. Selection Policy

As aforementioned, video captured from the CCTV camera is processed through a real-time background subtraction (BGS) module to detect the presence of motion activity in the sequence. Presence of motion activity for each frame of the sequence is

marked and recorded. This information is utilized by the motion estimation module of the encoder to perform selective motion estimation.

```
for (frame=1 to end of sequence)
  if (motion activity found)
    frameMotion [frame] = 1
  otherwise
    frameMotion [frame] = 0
end for

switch (motion estimation mode)
case GopByGop:
  for (each GOP of the sequence)
    for (first frame of GOP to GOP size)
      if ( frameMotion [frame] is 1)
        Perform Motion estimation for this GOP
        ME_performed = 1
        Break
    end for
    if ( ME_performed is not equal to 1)
        All the motion vectors are set to zero
  end for

case FrameByFrame:
  for (each frame of the sequence)
    if ( frameMotion [frame] is 1)
      Perform Motion estimation for this frame
   otherwise
        All the motion vectors are set to zero
  end for
```

Figure 50: Strategy for selective motion estimation

Two different selective motion estimation approaches, GOP level and Frame level, are implemented to improve the efficiency of the motion estimation process in terms of saving processing power and processing time. In the GOP level approach, BGS information is analysed for all the frames of a GOP. Therefore, a single decision of performing selective motion estimation is made for each GOP. If the BGS detects any moving object in any frame of the GOP then motion estimation is performed for that GOP otherwise motion vectors for all the frames of the GOP are set to zero. The workflow strategy of the proposed system is shown in Figure 50. GOP level selective motion estimation performs better when there is no motion activity for a large number of frames in the sequence. Its efficiency is lower when there is some pattern of activity present in the sequence not allowing to bypass the ME module. Also, this

approach is dependent on GOP size set for encoding the sequence where a smaller GOP size has better efficiency.

In the Frame level approach, the BGS information for each frame of the sequence is analyzed. The decision to perform selective motion estimation is made for each frame. If any motion activity is present in a particular frame then motion estimation is performed otherwise motion vectors for that frame are set to zero. This approach improves processing efficiency by performing the ME only for the frames where it is required and bypassing ME module for static frames. Thus, based on BGS analysis, no compromise is made for the frames which are important from a surveillance standpoint and complexity can still be reduced by applying the proposed approach.

6.3 Experimental Results for Selective ME

Performance evaluation of the proposed approach is carried out on different surveillance sequences given in Table 8. All of these sequences have CIF (352×288) spatial resolution and a frame rate of 30 Hz. Background in all the sequences is static throughout the length of the sequences.

While performing the experiment, the Sum of Absolute Difference (SAD) is used as a Block Distortion Measure (BDM). Block size is 16x16 while the search range is 15 (+- 15 pel displacement is possible in vertical and horizontal directions). All the videos are compressed into a 256 kbps bit-rate. True processing time is used to evaluate the performance of the proposed approach, while Y-PSNR is calculated to assess the image quality. The evaluation is performed using different GOP sizes. Each GOP contains at least one intra-coded frame. Thus, increasing the GOP size for the same sequence reduces the intra-coded frames in the whole compressed bit-stream. Consequently, a higher GOP size has a higher processing time. All the tests are performed on a machine with Intel Core(TM) 2CPU 6600@2.40GHz processor and 2 GB RAM. First of all, the BGS module has to be real-time to improve the efficiency of the proposed system. For this, Table 8 shows that the motion detection process is real-time where processing time for each surveillance sequence is given in seconds. BGS processes almost 30 frames in each second on the above described machine. Although BGS performance is real-time, still time consumed by BGS is included in overall encoding time for the evaluation of proposed selective motion estimation

approach. In all the tables, the PSNR results are in dBs and time is measured in seconds.

Seq	Total Frames	Time (sec)	Frames/Sec
Bridge	2100	66	31.82
Dance	500	16	31.25
Hall	300	10	30.00
LightOffOn	499	16	31.19
Parking	2100	68	30.88
RoadCar	499	15	33.26
Street	750	25	30.00
ThinkingMan	499	15	33.26

Table 8: Real-time motion detection

Experimental results for full search based motion estimation are summarized in Table 9. These results are used as a reference to compare the proposed approach. Table 10 shows the results for GOP level motion estimation. Different GOP sizes are selected to perform the experiment. With each GOP, the MCTF is performed in such a way to produce the maximum number of motion estimated frames.

Seq	GOP Size=8		GOP Size=16		GOP Size=32	
	Time(sec)	PSNR	Time(sec)	PSNR	Time(sec)	PSNR
Bridge	6336	40.04	8473	40.92	10396	41.33
Dance	1253	44.67	1652	46.43	2026	47.31
Hall	787	33.90	1045	36.54	1300	37.76
LightOffOn	1513	41.02	2067	42.67	2630	42.90
Parking	6080	37.15	8312	38.86	10582	39.61
RoadCar	1405	41.12	1917	43.07	2416	43.73
Street	1734	27.86	2309	30.96	2933	33.53
ThinkingMan	1186	30.03	1593	35.12	2005	39.36

Table 9: Full motion estimation

The processing time saving, compared to full motion estimation, achieved for GOP level approach is shown in Table 11. Results show that the nature of the sequence has great influence on the efficiency of the proposed approach. One drawback with GOP level motion estimation is that motion estimation is performed for all the frames of the GOP even if only one frame of the GOP has the motion activity. Thus to refine and improve the performance, Frame level selective motion estimation is implemented. Motion estimation is performed only for frames which contain any foreground object with some kind of motion activity. Table 12 and Table 13 show the experimental results for Frame level approach. Results show significant improvement over GOP level selective motion estimation approach.

Seq	**GOP Size=8**		**GOP Size=16**		**GOP Size=32**	
	Time(sec)	**PSNR**	**Time(sec)**	**PSNR**	**Time(sec)**	**PSNR**
Bridge	1795	40.04	3743	40.99	5172	41.38
Dance	786	44.63	1085	46.38	1490	47.31
Hall	764	33.91	1018	36.55	1310	37.76
LightOffOn	649	41.02	941	42.67	1363	42.90
Parking	2841	37.17	3790	38.87	4868	39.61
RoadCar	123	41.15	140	43.07	177	43.73
Street	1628	27.87	2232	30.96	2869	33.53
ThinkingMan	1157	30.04	1560	35.12	1914	39.36

Table 10: GOP level selective motion estimation

Seq	**GOP Size=8**	**GOP Size=16**	**GOP Size=32**
Bridge	71.67	55.82	50.25
Dance	37.27	34.32	26.46
Hall	2.92	2.58	-0.77
LightOffOn	57.11	54.48	48.17
Parking	53.27	54.40	54.00
RoadCar	91.25	92.70	92.67
Street	6.11	3.33	2.18
ThinkingMan	2.44	2.07	4.54

Table 11: Processing time saving for GOP level selective motion estimation (%)

Seq	GOP Size=8		GOP Size=16		GOP Size=32	
	Time(sec)	PSNR	Time(sec)	PSNR	Time(sec)	PSNR
Bridge	670	40.08	761	41.04	926	41.43
Dance	537	44.49	713	46.24	902	47.09
Hall	770	33.91	1033	36.55	1284	37.76
LightOffOn	600	41.84	800	42.72	1073	43.04
Parking	1454	37.17	1876	38.86	2372	39.59
RoadCar	123	41.15	144	43.07	174	43.73
Street	1228	27.87	1613	30.94	2010	33.50
ThinkingMan	282	30.20	339	35.33	407	39.47

Table 12: Frame level selective motion estimation

Seq	GOP Size=8	GOP Size=16	GOP Size=32
Bridge	89.43	91.02	91.09
Dance	57.14	56.84	55.48
Hall	2.16	1.15	1.23
LightOffOn	60.34	61.30	59.20
Parking	76.09	77.43	77.58
RoadCar	91.25	92.49	92.80
Street	29.18	30.14	31.47
ThinkingMan	76.22	78.72	79.70

Table 13: Processing time saving for Frame level selective motion estimation (%)

For assessing user perception based on visual quality, subjective quality evaluation based on the double stimulus impairment scale [69] method is performed as in Figure 51. Different users participated in this test. Videos from full motion estimation, GOP-by-GOP motion estimation and Frame-by-Frame motion estimation were organised at random. The user had to assign any number from 1 to 5 after watching the videos.

(a)

(b)

Figure 51: Visual comparison Hall frame 225 (a) Full ME (b) Frame level selective ME

Seq	Full ME	GOP-by-GOP	Frame-by-Frame
Dance	2.57	2.71	2.71
Hall	4.39	4.25	4.25
Street	2.82	2.68	2.53

Table 14: Subjective quality result

Table 14 shows the results for visual evaluation of the sequences. These are average numbers where 5 is the maximum number representing the best quality. Results show

that applying the proposed approach has not much effect on the visual perception of the video which is important from the surveillance standpoint. This shows that the processing efficiency for the proposed approach is improved without compromising on the visual quality of the surveillance videos.

6.4 Selective Block Search

In addition to the GOP level and Frame level selective motion estimation, Block level selective motion estimation is also implemented. This approach comes with some extra complexity because of identifying and locating non-static blocks. Once again, the BGS is used to identify static and non-static blocks. As shown in Figure 49, non-static pixels of each frame are bounded by the rectangular boxes. So, the location of these bounding boxes identifies non-static pixels. Motion vector (MV) search is performed only for those blocks which are identified by the BGS. Use of the BGS ensures that only the information sensitive from surveillance standpoint utilizes the computational resources. Motion vectors for block belonging to background are set to zero. The block selection for motion estimation is discussed in detail in the following section.

6.4.1 Block Selection Policy

As aforementioned, video captured from the CCTV camera is processed through a real-time background subtraction (BGS) module which forms rectangular boxes bounding the group of pixels which represent motion activity. Dimensions and coordinates of the bounding boxes are passed to the ME module of the video codec. Thus, based on BGS analysis, the bounding box indicates an area of the frame where ME needs to be performed. These indicated locations are important from surveillance standpoint. Workflow of the proposed system is shown in Figure 52. For each frame the ME module gets its required information from the BGS. If there are no pixels with motion in the frame then a motion vector for all the blocks of the frame is set to zero. If any motion activity is present in the frame then overlapping of bounding boxes and block under consideration is tested. In the case that there are any pixels common in block and bounding box then ME is performed for the block; otherwise,

the motion vector for the block is set to zero. The proposed approach is capable of handling multiple boxes for a frame.

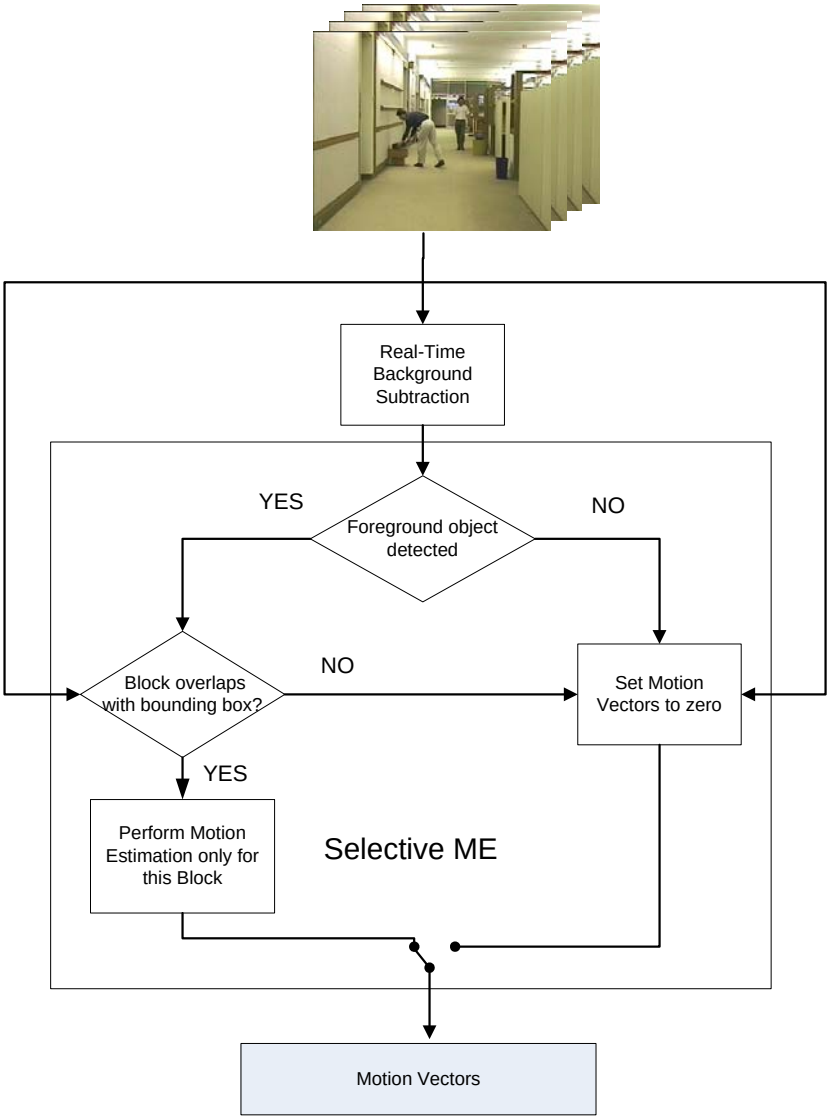


Figure 52: Workflow for selective motion estimation

To analyze the complexity of the proposed approach, let us assume that a frame is divided in *N* macroblocks (MBs) and time consumed for ME for each block is *T*. If there are '*X*' MBs which do not have any motion activity then conventional ME for the frame is presented as:

$$ME_{Conv} = \sum_{i=1}^{X} T_i + \sum_{j=X+1}^{N} T_j \qquad \textbf{(6.1)}$$

If a full search is employed then the time consumed by each MB is same. In this case, the conventional ME become:

$$ME_{Conv} = XT + (N - X)T \quad \textbf{(6.2)}$$

In the case of the proposed approach, only second term of the above equation is used.

$$ME_{Proposed} = (N - X)T + (N - X)T_b \quad \textbf{(6.3)}$$

where T_b is overhead cost of motion detection and block selection. Taking the ratio of $ME_{Proposed}$ to ME_{Conv} and after some mathematical manipulation we get the relation given below. This ratio describes how much time is consumed by the proposed approach compared to a conventional approach.

$$(ME_{Proposed})/(ME_{Conv}) = (N - X)(T + T_b)/NT \quad \textbf{(6.4)}$$

Clearly, if there is no motion activity in the frame i.e. X = N then the time consumed by the proposed approach becomes zero and for the worst case when all the blocks of the frame have motion activity i.e. X = 0 then the proposed approach consumes T_b/T more time compared to FS.

6.5 Experimental Results for Selective Block ME

The performance of the proposed framework has been evaluated using different typical surveillance sequences as described in section 8.3. Different criteria are used for evaluating the proposed selective search. They are the Mean Square Error (MSE), average number of search points per motion vector (SearchPt), Relative Complexity Reduction (RCR), the SpeedUp ratio, the Y-PSNR value, and true processing time. Any of the available block matching algorithms can be utilised to perform a search. In this test, the search technique is based on Diamond Search (DS) and the full search (FS) algorithms.

Experimental results for selective block and all blocks motion vector search are summarized in Table 15, Table 16 and Table 17.

Seq	GOP Size=8		GOP Size=16		GOP Size=32	
	Time(sec)	PSNR	Time(sec)	PSNR	Time(sec)	PSNR
Bridge	743	40.08	838	41.04	1038	41.43
Dance	139	44.37	153	46.04	184	46.83
Hall	123	33.91	137	36.36	167	37.57
LightOffOn	187	41.83	210	42.07	254	42.98
LightOnOff	194	34.01	220	37.55	271	39.42
Parking	716	37.15	814	38.83	988	39.57
RoadCar	143	41.18	165	43.10	203	43.75
ThinkingMan	171	30.22	201	35.36	241	39.49

Table 15: Block level selective motion estimation

Seq	GOP Size=8	GOP Size=16	GOP Size=32
Bridge	88.27	90.11	90.02
Dance	88.91	90.74	90.92
Hall	84.37	86.89	87.15
LightOffOn	87.64	89.84	90.34
LightOnOff	84.35	86.72	86.93
Parking	88.22	90.21	90.66
RoadCar	89.82	91.39	91.60
ThinkingMan	85.58	87.38	87.98

Table 16: Processing time saving for Block Level selective motion estimation(%)

Sequence	SDS		DS	
	MSE	SearchPt	MSE	SearchPt
Hall	8.25	1.84	8.50	17.32
Dance	2.34	1.78	2.45	15.30
Street	22.43	0.53	24.37	14.93

Table 17: Performance comparison using diamond search

Table 18 shows the computational power saving for each sequence in terms of the Relative Complexity Reduction (RCR) and the SpeedUp ratio. RCR and SpeedUp ratio are calculated using the following relationships:

$$RCR = (1 - NS_{Sel} / NS_{Com}) \times 100\ \% \quad \textbf{(6.5)}$$

$$SpeedUp = NS_{Com} / NS_{Sel} \quad \textbf{(6.6)}$$

where NS_{Sel} and NS_{Com} represent the total number of search points using the proposed selective search approach and complete search approach, respectively.

Sequence	NS_{Sel} | NS_{Com}	***RCR* (%)**	***SpeedUp***
Hall	429971 | 4061538	89.41	9.45
Dance	692493 | 5962671	88.39	8.61
Street	307007 | 8725698	96.48	28.42

Table 18: Complexity reduction using diamond search

The experimental results of Table 17 and Table 18 show that the proposed approach can reduce the computational complexity significantly with very little change in MSE. These parameters are the performance indicators for the BMA only. The overall encoding time is affected by the performance of the BMA.

Sequence	**SDS**	**DS**	**SFS**	**FS**
Hall	39.15	38.84	39.15	38.96
Dance	44.76	44.23	44.87	44.72
Street	35.06	34.26	35.06	34.79

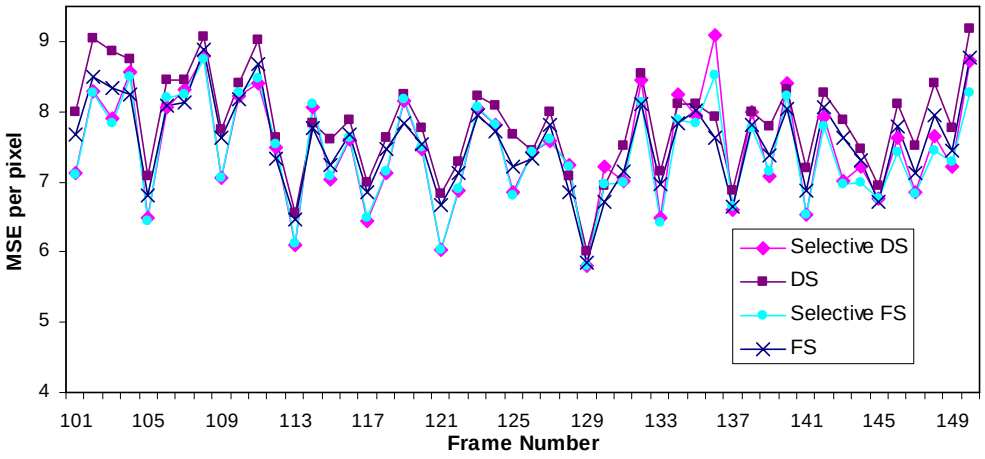
Table 19: PSNR results for diamond and full search BMA

Y-PSNR results for selective diamond search (SDS), diamond search (DS), selective full search (SFS) and full search (FS) are shown in Table 19. In SFS, a full search is performed for blocks which are bounded by boxes. Some of PSNR and MSE results are better for the proposed selective search. This is because of setting most of the motion vectors (MV) to zero which saves bits from MV encoding process. These saved bits are utilized by rate-distortion module of the encoder to improve quality of image.

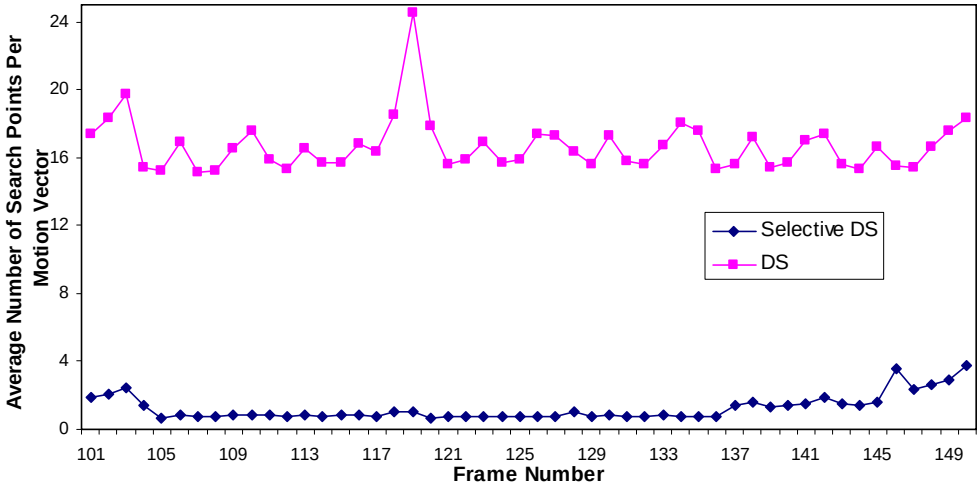
The true processing time of the whole encoding system is given in Table 20. Encoding parameters perform 5 temporal decomposition levels which produce plenty of motion estimated frames. Table 20 shows a significant time reduction for the proposed approach even applying a full search BMA for the selected blocks.

Sequence	**Proposed System** (Sec)			**FS** (Sec)	**%age saving**
	BGS	**SFS**	**Total**		
Hall	10	216	226	385	41.30
Dance	17	256	273	432	36.81
Street	25	500	525	823	36.21

Table 20: True processing time using full search BMA

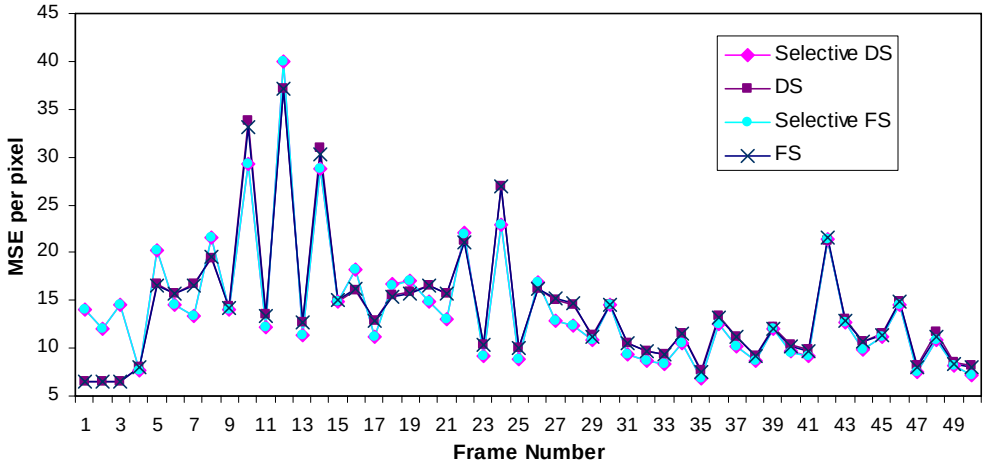


(a) MSE

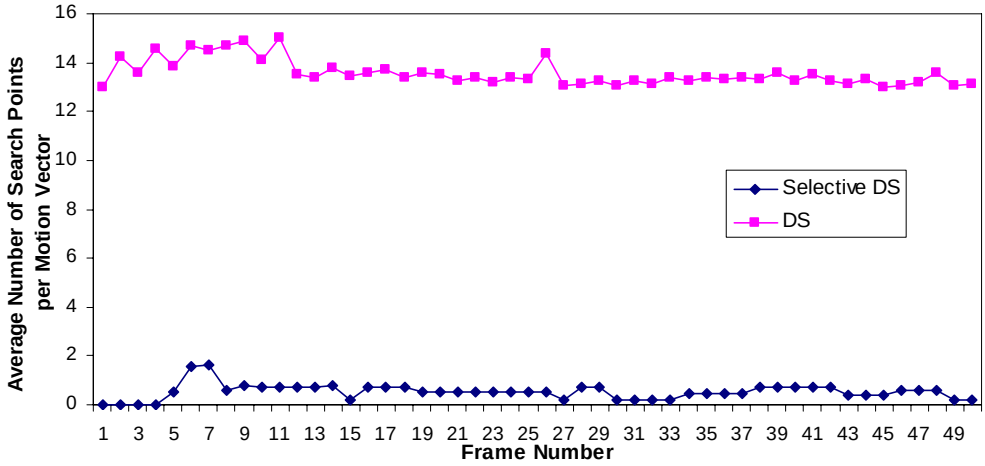


(b) Search points

Figure 53: Test results for Hall video, frame 101- 150



(a) MSE



(b) Search points

Figure 54: Test results for first 50 frames of Street video

Figure 53 illustrates a frame-by-frame performance comparison of the proposed approach against a conventional search approach for hall videos. For the hall video, frame 101-150 are shown due to a high motion activity in these frames. Results show that the proposed approach can reduce the complexity of ME while maintaining the visual quality.

6.6 Tracker Based

Video tracking applications are used to facilitate the visual surveillance. Whenever an object moves in the surveillance sequence, the video tracker starts performing motion

calculations for the object. The object itself is bounded by a rectangular box to make the moving object prominent in the video. In proposed tracker based motion estimation, a fast surveillance video tracking approach based on the work of Stauffer and Grimson [70] is used. The video tracker assigns each moving object a unique track number. These track numbers or tracks Ids are matched to identify same object in different frames of the sequence. A simple workflow for the tracker based motion estimation is shown in Figure 55.

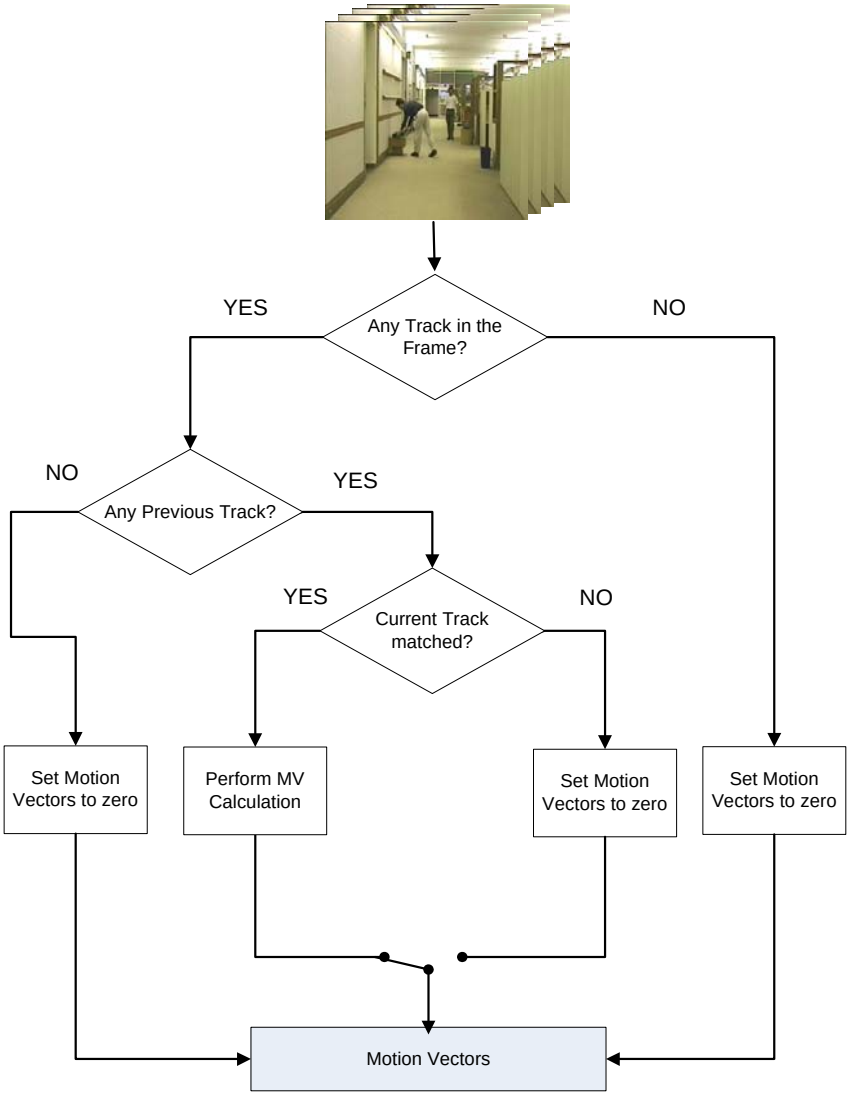


Figure 55: Workflow for tracker based motion estimation

In the process of motion estimation, each block of the frame is a candidate for motion vector calculations. So, the very first step is to check whether the frame for motion

estimation has any object with any tracking information. If there is no track present then the motion vectors for all the blocks of the frame are set to zero. This is because the background in the surveillance videos remains static through out the sequence. So, zero motion vectors retain the same block from the previous frame.

Now, if the candidate frame has some moving objects with some tracking information then each block of the frame is tested as to whether it is part of the tracked objects. If a block is not part of any tracked object then it indicates that the block represents the background of the sequence. Therefore, the motion vectors for such blocks are set to zero. If a block is part of any of the tracked objects present in the frame then the further tests are performed for motion vector calculation. First of all, the track number or track Id of the candidate block is matched against the track numbers of the previous frame; if the track number is found then the displacement of the bounding rectangular box between two frames is taken as a motion vector. This displacement can be calculated by taking the difference between any of the corner points of the bounding boxes.

The tracker based approach is faster because there is no need to perform any block matching technique within a search window. This helps in reducing the complexity of the motion estimation. On the other hand, this approach may reduce the visual quality of the tracked object because all the blocks of the tracked object are assigned the motion vector value. This quality impairment can be observed in a situation where a tracked object is moving in one direction but some parts of the object, let us say arms in the case of human, are moving in different direction with different displacement and/or different pose.

6.7 Experimental Results for Tracker Based ME

Results for the tracker based motion estimation are summarized in Table 21 and Table 22. The comparison of selective and tracker based motion estimation in terms of time saving and image quality is presented in Figure 56 to Figure 59. The processing time saving, compared to full motion estimation, achieved for the tracker based motion estimation approach is shown in Table 22. Results show that the nature of the sequence has great influence on the efficiency of the proposed approach where level and type of motion activity are the main factors. For sequences like ‘Bridge’

which contain very little motion activity, tracker based motion estimation does not show any improvement over frame based selective motion estimation.

Seq	GOP Size=8		GOP Size=16		GOP Size=32	
	Time(sec)	PSNR	Time(sec)	PSNR	Time(sec)	PSNR
Bridge	1194	39.97	1325	40.96	1525	41.36
Hall	208	31.95	233	33.93	259	35.03
LightOffOn	312	41.66	351	42.49	398	42.90
Parking	1187	36.92	1327	38.57	1535	39.33
RoadCar	230	41.09	255	43.02	292	43.69
Street	473	27.5	522	30.17	594	32.37
ThinkingMan	283	29.92	315	34.75	363	38.79

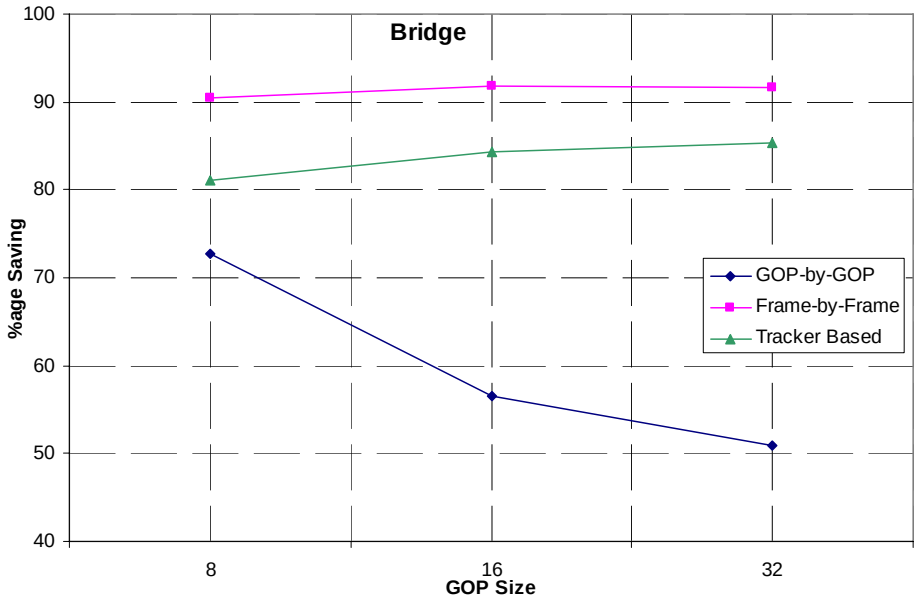
Table 21: Tracker-Based motion estimation

Seq	GOP Size=8	GOP Size=16	GOP Size=32
Bridge	81.16	84.36	85.33
Hall	73.57	77.70	80.08
LightOffOn	79.38	83.02	84.87
Parking	80.48	84.04	85.49
RoadCar	83.63	86.70	87.91
Street	72.72	77.39	79.75
ThinkingMan	76.14	80.23	81.90

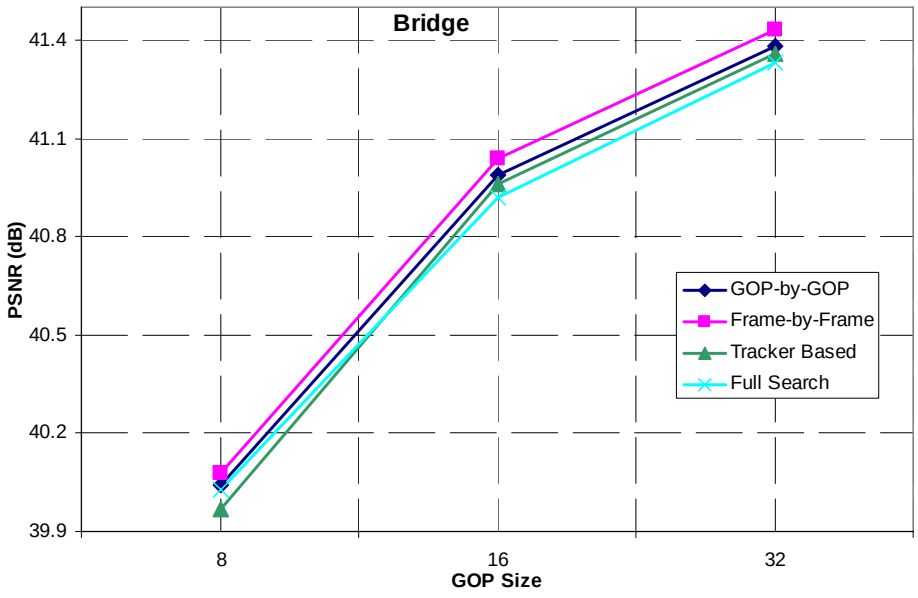
Table 22: Processing time saving for Tracker-based selective ME (%)

On the other hand, for sequences like ‘Hall’ and ‘Street’ which contain a high level of motion activity, tracker based motion estimation speeds up the motion estimation process but the visual quality of the decoded sequence is degraded. The reason for this quality degradation is the type of motion activity carried out by the foreground object. In these sequences, many of the pixels do not have same direction of movement as the direction of the main foreground object linked to these pixels. So, the motion vectors for the blocks which hold such pixels do not point to the right pixels in the reference frame while decoding. Consequently, the visual quality of the

reconstructed sequence is degraded. Results show a significant improvement over the GOP level selective motion estimation approach.

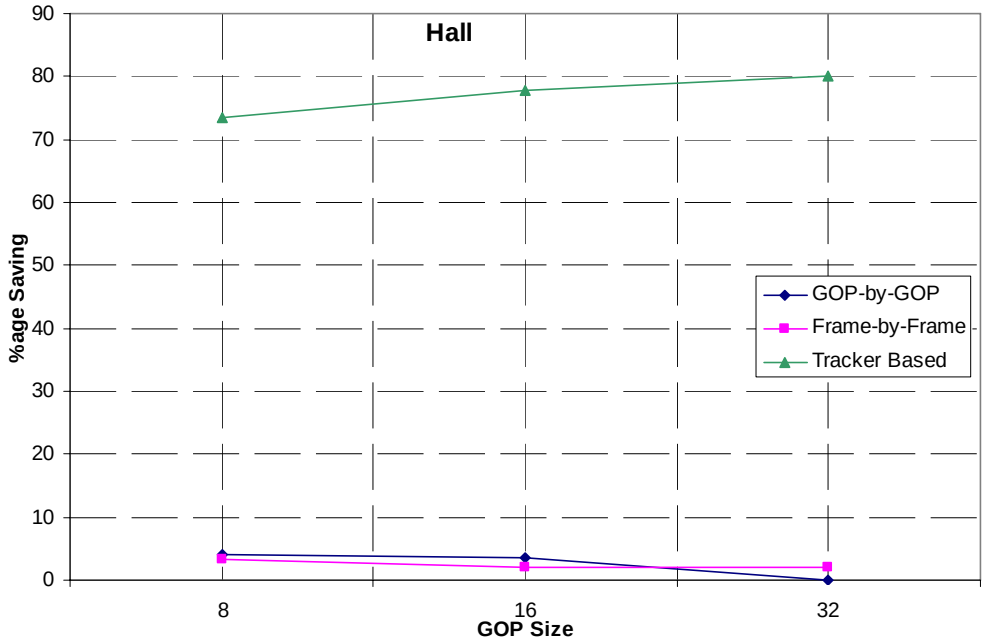


(a)

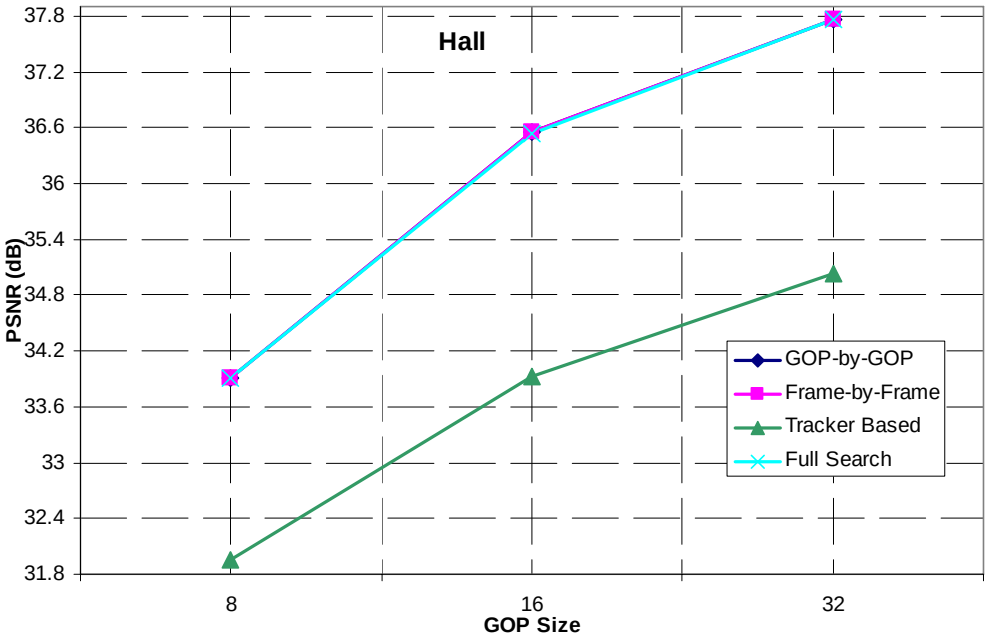


(b)

Figure 56: Bridge sequence comparison of tracker based and selective motion estimation (a) %age time saving (b) visual quality

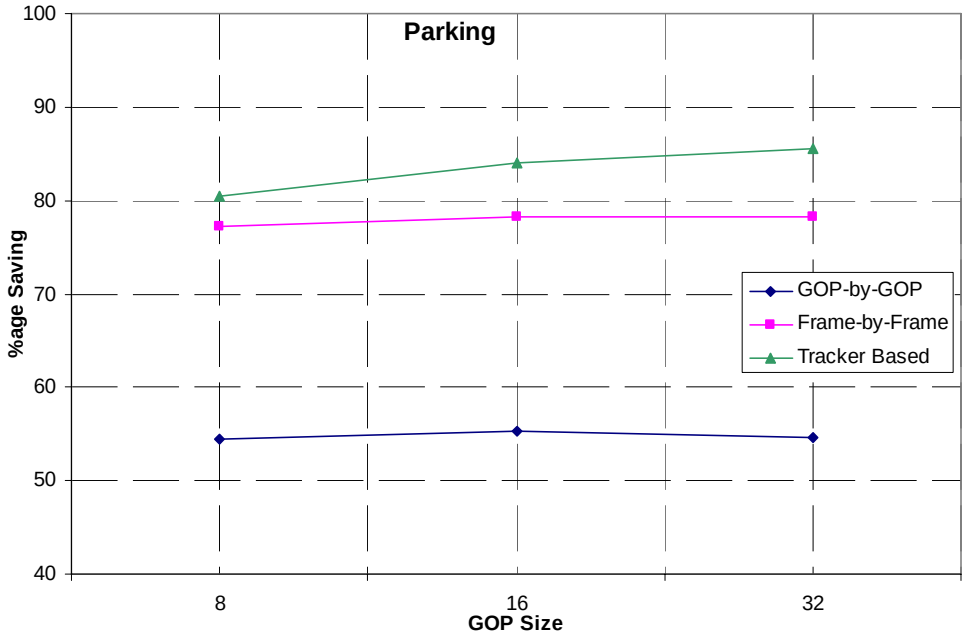


(a)

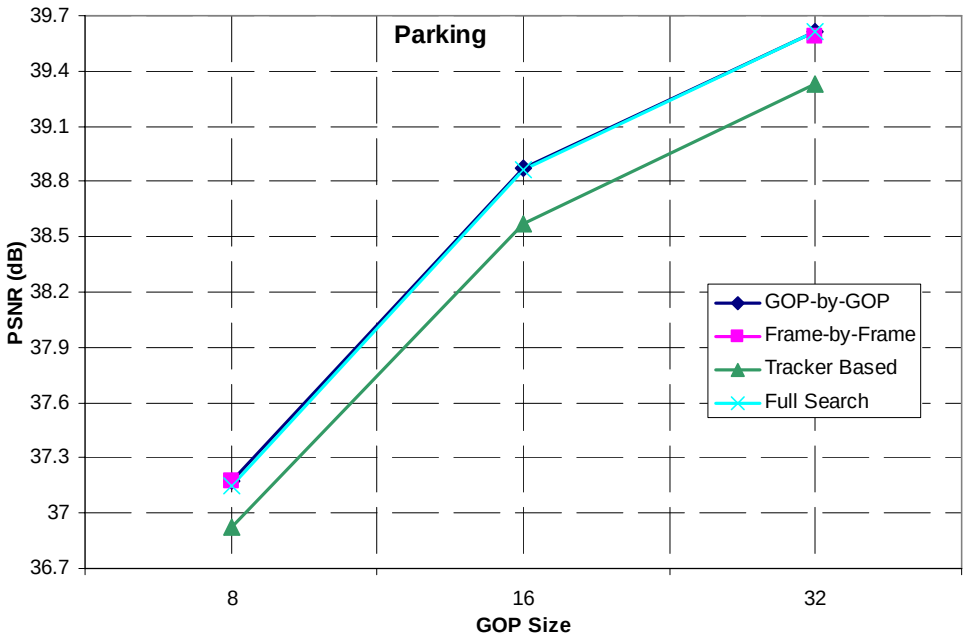


(b)

Figure 57: Hall sequence comparison of tracker based and selective motion estimation (a) %age time saving (b) visual quality

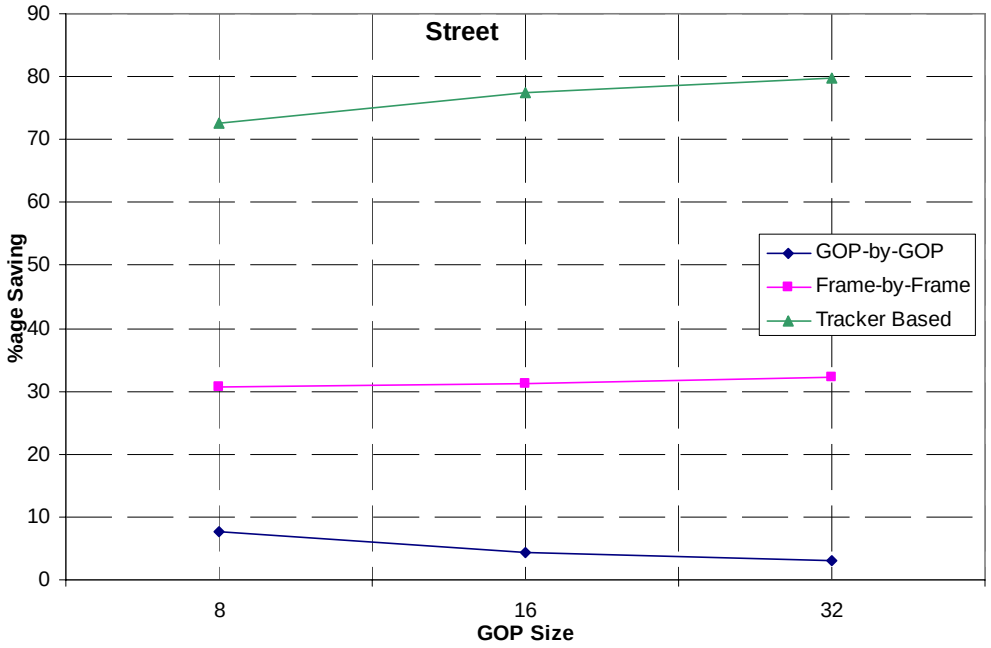


(a)

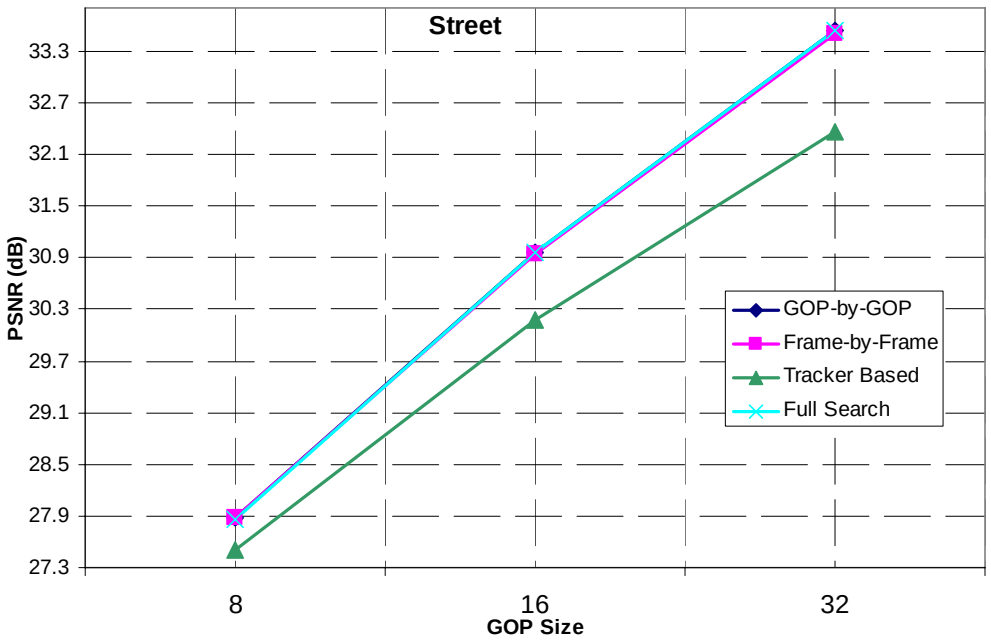


(b)

Figure 58: Parking sequence comparison of tracker based and selective motion estimation (a) %age time saving (b) visual quality



(a)



(b)

Figure 59: Street sequence comparison of tracker based and selective motion estimation (a) %age time saving (b) visual quality

6.8 Conclusion

A novel approach for performing selective motion estimation was proposed where object detection information generated by the VCA was used to flag the frames which did not contain any moving object. Based on this analysis, different selective motion estimation approaches were proposed which included: GOP level selective motion estimation, Frame level selective motion estimation and block level motion estimation.

In the GOP level selective motion estimation, the decision as to whether perform motion estimation or to skip was enforced at the GOP level. Therefore, the only way of skipping the motion estimation for a particular GOP was the scenario where there was not a single frame of the GOP identified as containing a moving object. Due to the very low probability of such a scenario, this scheme had little improvement even for moderately busy locations. The second drawback of the GOP level selective motion estimation was its dependency on the GOP size. So, with a smaller GOP size, there was higher probability of occurring such GOPs which do not have any frame detected containing a moving object. To counter the issues faced in GOP level selective ME, a Frame level selective ME was proposed where decision of performing ME or skipping it was taken for each frame independent of other frames. Once again, this approach was integrated in the SCC framework keeping the contact with the SCC framework. Evidently, the Frame level selective ME performed better than the GOP level. The key observation for Frame level ME was the imposing of ME decision for all the blocks of the frame irrespective of the location and size of the moving objects.

Under the motivation of evolving selective ME from Frame to macroblock level, Block level selective ME was proposed. Two major challenges to implement this approach were (i) identification of the macroblock as part of a moving object and (ii) locating those blocks while performing ME. As explained in Chapter 5, the foreground pixels identified by the VCA were isolated from the background pixels by using the information in the rectangular bounding boxes. A similar solution to address these two issues of Block level selective ME was used for the implementation.

After the paradigm of selective motion estimation, a novel way of performing efficient motion estimation through reusing the information of surveillance video

object tracker was proposed. In this approach, a real-time object tracker was used which generates information for each unique object with a unique track identity. In addition to this, objects were bounded in a rectangular box. So, instead of performing any kind of motion estimation for any block of the surveillance video, the motion vectors are calculated through the information generated by the object tracker. This approach had the drawback of miscalculating some of the motion vectors corresponding to the same object; ultimately, reducing the visual quality.

Chapter 7

Fast Full Search For Motion Estimation

Fast motion vector search algorithms have been developed under the motivation of reducing the computational complexity of the exhaustive full search. All of these algorithms are based on monotonically increasing match criteria around the location of the optimal motion vector to iteratively determine that location. The computational complexity is reduced by selecting a limited number of test points within the search window. These search algorithms follow the route of local minima to find out the ultimate motion vector for a particular search. This can lead to reduction in the accuracy of the motion estimation and ultimately the quality of video. Thus, reduction in computational complexity comes at the cost of reduction in accuracy. To address this problem, fast full search algorithms ([76] - [83]) have been proposed.

.

7.1 Successive Elimination Algorithm

Successive elimination algorithm (SEA) [76] constrains the search process within the search window while preserving the optimal solution for the motion vector. SEA is based on mathematical inequality given below.

$$R - M(x, y) \leq SAD\ (x, y) \quad \textbf{(7.1)}$$

$$M(x, y) - R \leq SAD\ (x, y) \quad \textbf{(7.2)}$$

In these equations, R represents the some norm of the reference block while *M(x, y)* represents the sum norm of any matching candidate block with the motion vector *(x, y)*. The term on the right side is the sum of the absolute difference (SAD) between the reference block and the candidate block with motion vector *(x, y)*. Now, assume for a given *SAD (m, n)* for an initial matching candidate block with motion vector *(m, n)*. Obviously, the potential candidate for the best match is that block for which the following equation holds.

$$SAD\ (x, y) \leq SAD\ (m, n) \quad \textbf{(7.3)}$$

Then equations 7.1 and 7.2 can be modified as follows.

$$R - M(x, y) \leq SAD\ (m, n) \quad \textbf{(7.4)}$$

$$M(x, y) - R \leq SAD\ (m, n) \quad \textbf{(7.5)}$$

which implies

$$R - SAD\ (m, n) \leq M(x, y) \leq R + SAD\ (m, n) \quad \textbf{(7.6)}$$

The inequality in equation 7.6 is the major result used by the SEA search. It indicates that to obtain the best match, only those blocks are tested in the search process whose sum norms satisfy the equation 7.6. The set of these blocks is less than the total number of blocks in a search window. So, utilizing equation 7.6 helps to reduce computational complexity of the search process greatly without excluding the

optimum point. An efficient method to calculate the sum norms for candidate blocks in the search window is given in the work of Li and Salari [76].

7.2 Fast Full Search in H.264

Advance video coding (AVC) also known as H.264/MPEG-4 Part 10 [24] [25] has been developed by the joint video team (JVT). One of the novelties contributing to the superior performance of H.264 is a rich set of coding modes to choose from for each macroblock. One of them is the ability to select different block partition (16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4) for better motion estimation. For a full search approach, each partition is selected to perform matching within the search window. The use of different block sizes for various macroblock partitions multiplies the number of computations. So, the improved accuracy in the motion estimation comes with a substantial increase in the computational complexity.

A fast full search approach is given in the work presented by Ates and Altanbasak [86] to reduce the higher complexity introduced by different macroblock partitions. The fast full search is based on the fact that, if the same search range is used for all block sizes, the sum of absolute differences (SADs) computed for small block partitions (e.g. 4×4) can be reused to construct the SADs for larger block partitions (e.g. 16×16). That is, for a given motion vector, SADs calculated for 4×4 sub-blocks can be added up to find the SADs of larger sub-blocks. This will require additional memory to store the computed SADs, but decrease the number of computations almost to the level of a single block type case.

Let us assume that *Bmxn* is a sub-block with size *(m x n)*, such that *(m x n)* $\in$ {(16×16), (16×8), (8×16), (8×8), (8×4), (4×8), (4×4)}, then SAD for *Bmxn* is given by the following equation [86].

$$\mathrm{SAD}_{Bmxn}(\mathrm{v}) = \sum_{x=1,y=1}^{m,n} \left| c(x,y) - r(x+v_x, y+v_y) \right| \qquad \textbf{(7.7)}$$

where $v = (v_x, v_y)$ is the motion vector, 'c' and 'r' are the current and reference frames, respectively. As the SAD reuse is based on the simple observation that, for a given

MV $v = (v_x, v_y)$ SAD of *Bmxn* can be decomposed into the SADs of its 4×4 sub-blocks as given below [86].

$$\mathrm{SAD}_{Bmxn}(\mathrm{v}) = \sum_{\substack{k=1 \\ l=1}}^{m/4,n/4} \sum_{\substack{x=4k-3 \\ y=4l-3}}^{4k,4l} \left| c(x,y) - r(x+v_x, y+v_y) \right| \qquad \textbf{(7.8)}$$

This means, to compute $\mathrm{SAD}_{Bmxn}(\mathrm{v})$ for a sub-block *Bmxn* with MV $v = (vx, vy)$ it is necessary to compute SADs of all its 4×4 sub-blocks such that (4x4) $\subset$ *Bmxn*. Thus from equation 7.8, it is concluded that once SADs for all the 4×4 sub-blocks are computed, they can be reused to build up SADs of sub-blocks with higher dimensions. The computational complexity for performing search for all the seven macroblock partitions becomes close to the computational complexity of searching for only 4×4 sub-blocks. This reduction in computational complexity comes with a higher use of memory where SAD values for all the partitions are stored to build SAD values for higher dimension partitions

SAD^1_{4x4}	SAD^2_{4x4}	SAD^3_{4x4}	SAD^4_{4x4}
SAD^5_{4x4}	SAD^6_{4x4}	SAD^7_{4x4}	SAD^8_{4x4}
SAD^9_{4x4}	SAD^{10}_{4x4}	SAD^{11}_{4x4}	SAD^{12}_{4x4}
SAD^{13}_{4x4}	SAD^{14}_{4x4}	SAD^{15}_{4x4}	SAD^{16}_{4x4}

Table 23: SAD values for single search position in search window

Table 23 shows calculated SAD values for a single search position in the search window for a block of 16x16. Let us assume search range is '*r*' then search window will consist of *(2r + 1) x (2r + 1)* search positions. For each search position, there will a table similar to Table 23. These SAD values are reused to build similar tables

for higher dimension sub-block partitions containing SAD values for that particular sub-partition. Some examples of SAD reuse are shown below.

$$SAD^{1}_{8x4} = SAD^{1}_{4x4} + SAD^{2}_{4x4}$$
$$SAD^{1}_{4x8} = SAD^{1}_{4x4} + SAD^{5}_{4x4}$$
$$SAD^{1}_{8x8} = SAD^{1}_{4x8} + SAD^{2}_{4x8}$$
$$SAD^{1}_{16x8} = SAD^{1}_{8x8} + SAD^{3}_{8x8}$$
$$SAD^{1}_{8x16} = SAD^{1}_{8x8} + SAD^{2}_{8x8}$$
$$SAD^{1}_{16x16} = SAD^{1}_{8x16} + SAD^{2}_{8x16}$$

7.3 Multiple Reference Frames Motion Estimation

The AVC/H.264 codec uses multiple reference frames to enhance the accuracy of motion estimation. In the multiple reference frame motion estimation, a single block with uni-prediction in *P* slices is predicted from one reference picture out of a large number of decoded pictures. With a similar approach, a biprediceted block in *B* slices is predicted from two reference pictures; both can be chosen from their candidate reference picture lists. Multiple reference frame motion estimation is an effective technique to improve the coding efficiency. However, this approach dramatically increases the computational complexity of the encoders because the motion estimation process needs to be performed for each of the reference frames. So, the computational complexity is increased ‘n’ times if ‘n’ number of reference frames are used. The work presented in this chapter is focused on the reduction of computational complexity for multiple reference frames with a quality similar to that achieved through the full search.

7.4 Fast Multiple Reference Motion Estimation

Visual quality is one of the most important factors for the surveillance videos. For this reason, a full search based approach with less computational complexity is required for fast and accurate motion estimation. At the same time, a multiple reference based approach is adopted because of its more accurate results for motion

estimation. Successive Elimination Algorithm (SEA) described in section 7.1 is investigated for multiple reference frames based fast full search.

7.4.1 The Algorithm

Assuming that there are five reference frames for a motion estimation of a 'P' slice as depicted in Figure 60. A difference frame is generated for each pair of consecutive reference frames. Ideally, two consecutive frames are exactly the same except for those pixels which are changed due to the motion of an object.

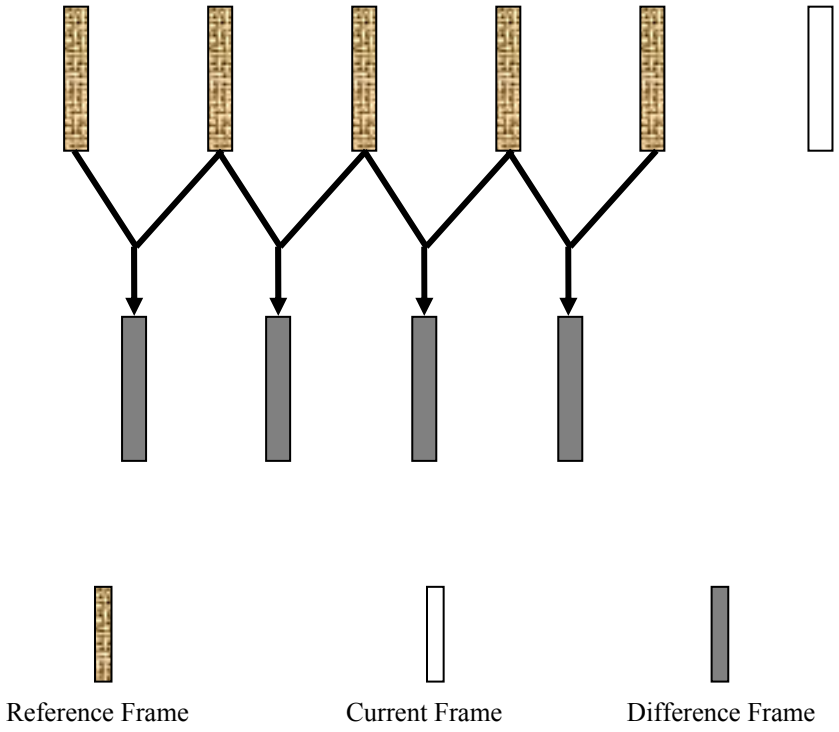
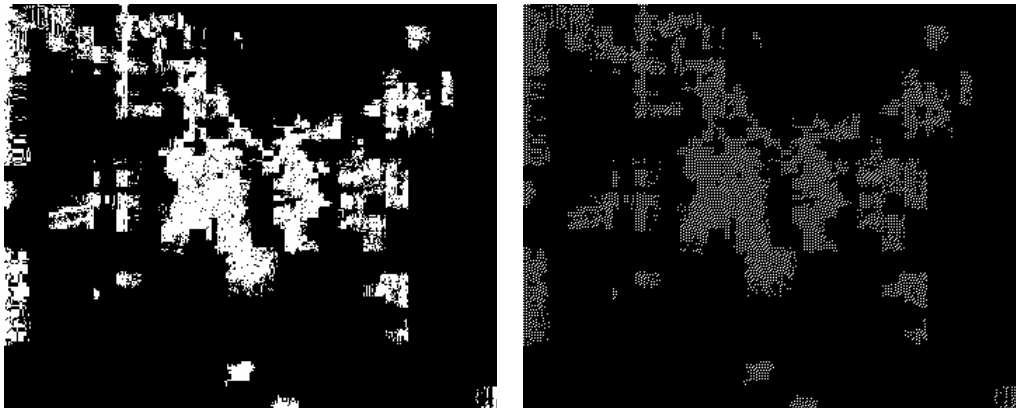


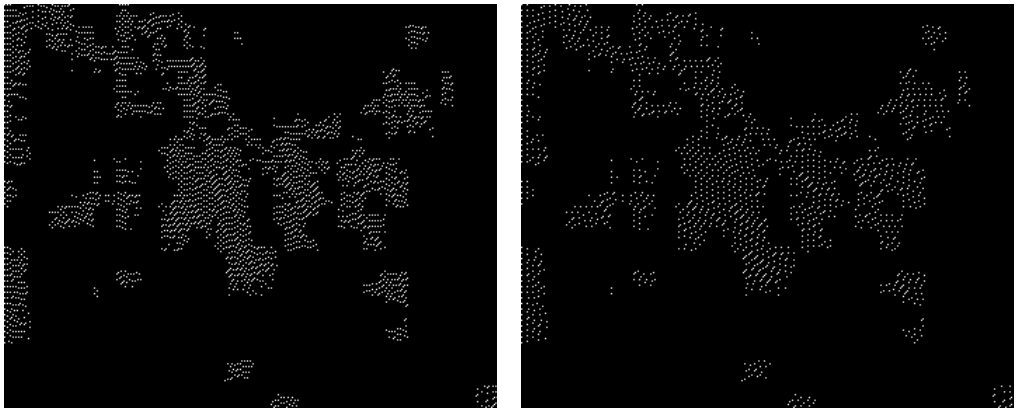
Figure 60: Multiple reference frames with difference frames

Thus, the difference frame contains all the pixel values equal to zero except the pixels which represent motion. So, the difference frames are the indicator of the locations for object motion. Practically, two consecutive frames have different pixel values even for the locations where there is no motion at all. This is because of environmental noise, camera noise and noise introduced by wind or flickering lights, etc. This is shown in Figure 61 for two consecutive reference frames of the Hall sequence with the difference frame in (c).

(a) 1st Reference frame (b) 2nd Reference frame



(c) Difference frame (no grouping) (d) Difference frame (2x2 grouping)



(e) Difference frame (2x4 grouping) (f) Difference frame (4x4 grouping)

Figure 61: Consecutive reference frames with difference frame

For the given fast multiple reference motion estimation, the main idea is to perform the full motion vector search in the reference frame which is the nearest to the current

frame. This full search is based on the successive elimination algorithm [76]. For the rest of the reference frames, difference frames are used to indicate the locations representing the change in the pixel value. So, only those points are tested as motion vector candidates which are indicated by the difference frames. This approach is equivalent to performing the full search because all the similar points in two consecutive frames are already tested in the previous reference frame.

The difference frame consists of so many different points close to each other as shown in Figure 61 (c). These points can be grouped together represented by a single point in the group. The difference frames for group 2×2, 2×4 and 4×4 are shown in Figure 61 (d), (e), (f). Reduction in the density of the difference pixel localization can be seen with the increase in the grouping dimensions. A minimum block size for the motion vector search is 4×4 in the H.264. Thus, all these groupings for the difference frames are covered by 4×4 search blocks with an extra search around the optimum point.

Multiple Reference Frame Motion Estimation (MRFME) for improved processing efficiency has been categorized into three implementation approaches, (i) MRFME_1 (ii) MRFME_2 and (iii) MRFME_3. These approaches are described in Table 24.

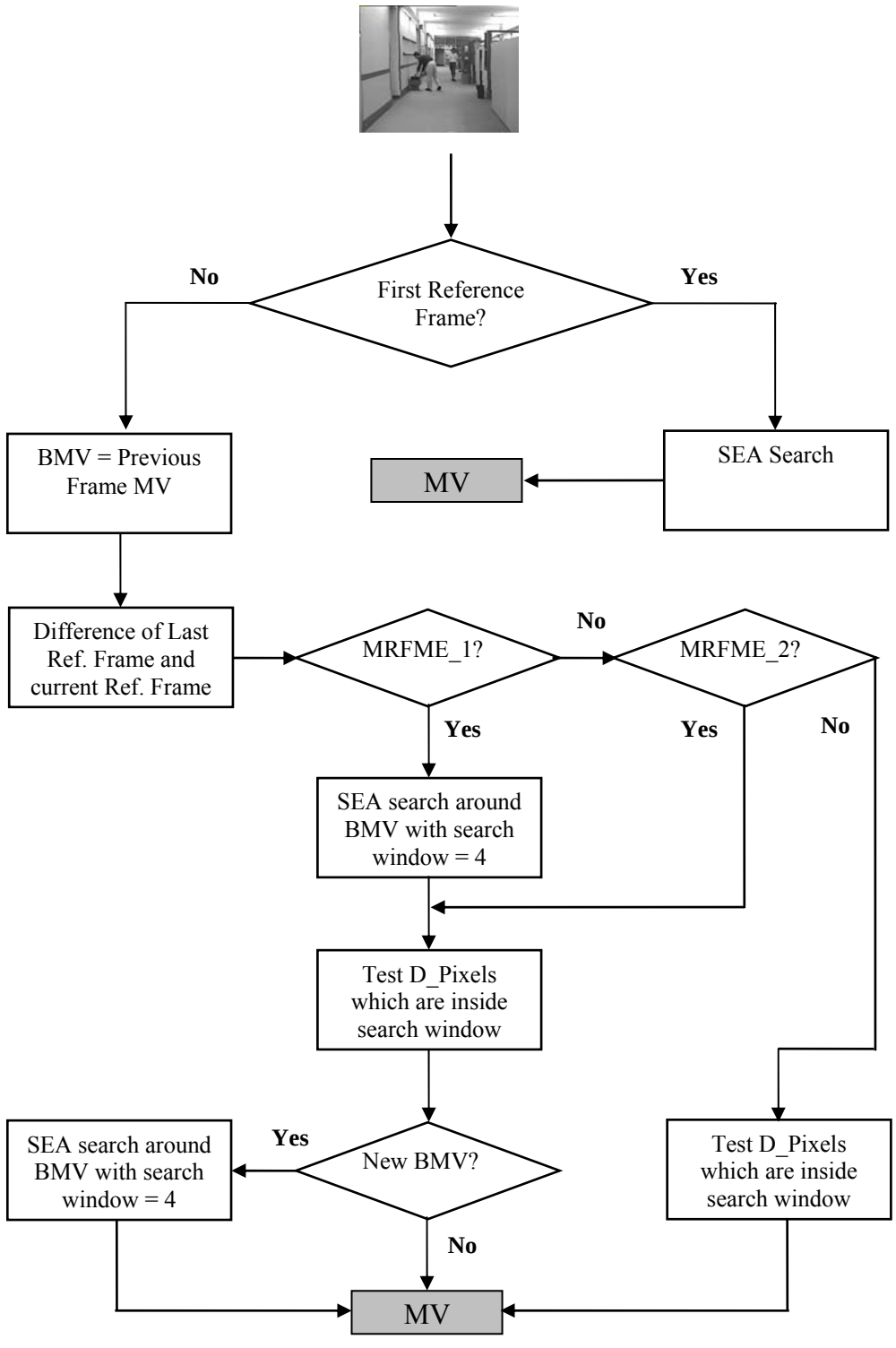
MRFME_1 performs SEA based motion estimation around previous frame motion estimation with a smaller search window. This step is based on the observation that the majority of motion vectors are close to the previous frame/optimum motion vector. Thus, by performing an initial SEA search non-candidate points are eliminated. After checking the test points indicated by the difference frames, once again, the SEA search is performed to refine the MV and to cover up the grouping of the pixels. The MRFME_2 is exactly same as the MRFME_1 without the initial SEA search while the MRFME_3 performs testing of only those points which represents the group of pixels.

The flowchart diagram for the implemented algorithm is given in Figure 62. Each reference frame is tested for being the first reference frame for the motion vector search. If it is the first reference frame then SEA search is performed to find the best MV. For the rest of the reference frames, MRFME_1, MRFME_2 and MRFME_3 approaches are used to find better motion vector. The terms BMV and D_Pixels in the flowchart diagram represent the best updated motion vector and different pixels from

the last reference frame, respectively. D_Pixels are actually group representative locations for 2×2, 2×4 or 4×4 groups as described previously.

Search type	Description
MRFME_1	(i) Best MV = Previous frame MV (ii) SEA search is performed around Best MV with search window = 4. If better MV is found then Best MV is replaced with it. (iii) All the different pixel group locations within the original search window are tested for better MV. If better MV is found then to cover different pixels in the group, SEA search is performed around the found location with search window size equal to 4.
MRFME _2	(i) Best MV = Previous frame MV (ii) All the different pixel group locations within the original search window are tested for a better MV. If a better MV is found then to cover different pixels in the group, the SEA search is performed around the found location with search window size equal to 4.
MRFME _3	(i) Best MV = Previous frame MV (ii) All the different pixel group locations within the original search window are tested for better MV. If better MV is found then this location is selected as optimum MV.

Table 24: Search approaches for reference frames except first reference frame



BMV = Best MV D_Pixels = Different Pixels from last ref. frame

Figure 62: Flowchart for the implemented algorithm

7.5 Experimental Results

The performance evaluation of the proposed approach is carried out on five different surveillance sequences: 'Bridge', 'Dance', 'Hall', 'CarPark' and 'Street' as shown in Table 25. All of these sequences have CIF (352×288) spatial resolution and a frame rate of 30 Hz. Background in all the sequences is static throughout the length of the sequences. "Bridge" sequence is a distant view of a boat moving in a river. The 'Dance' sequence contains an animated person dancing with fast legs and arms motion. In the 'Hall', two persons are walking in opposite directions in a corridor. In the 'CarPark', a car comes in a parking area. In the 'Street', with real outdoor street background, different animated objects move through the street.

Seq.	Total frames	Description
Bridge	500	Boat moving slowly
Dance	500	Animated man with fast legs and arms movement
Hall	300	Persons walking in a corridor
CarPark	500	Car in a parking area
Street	750	Different objects moving in a street

Table 25: Surveillance sequences

While performing the experiment, the Sum of Absolute Difference (SAD) is used as a Block Distortion Measure (BDM). The proposed algorithm is implemented in the reference software, Joint Model (JM). The proposed algorithm was implemented after introducing major changes in the motion estimation modules and reflecting those changes throughout the software. The main profile of H.264 codec is used. As the process of multiple reference frames is the same for both 'B' and 'P' frames, therefore only 'P' frames are used in the test. The block size is 16x16 while the search range is 16 (+- 16 pel displacement is possible in vertical and horizontal directions from the current block location). The maximum number of reference frames is 5. All the videos are compressed for 196 kbps bit-rate. The true processing time is used to evaluate the performance of the proposed approach, while Y-PSNR is

calculated to assess the image quality. All the tests are performed on a machine with Intel Core(TM) 2CPU 6600@2.40GHz processor and 2 GB RAM.

A comparison of the full search, fast full search and SEA search is given in Table 26. This comparison is based on motion estimation time only. It can be seen that the SEA search has performed better in terms of motion estimation time with similar image quality. In the implementation of the SEA search algorithm, memory address calculation is performed only for the first block of the search window then a sequential increment is used to locate the remaining block in the search window of the reference frame. This helps to speed up the memory access process and thus improves the motion estimation process. Consequently, the implemented approaches of the MRFME are compared with the SEA search results.

Seq.	**Full Search**		**Fast Full Search**		**SEA**	
	T(sec)	**YPSNR**	**T(sec)**	**YPSNR**	**T(sec)**	**YPSNR**
Bridge	2732	39.31	1298	39.29	512	39.32
Dance	2730	45.78	1056	45.78	149	45.8
Hall	1673	37.19	634	37.19	268	37.18
CarPark	2671	41.6	1061	41.6	731	41.61
Street	3457	33.95	1588	33.96	1011	33.93

Table 26: Comparison of full search approaches

As aforementioned, difference frame pixels groups of 2×2, 2×4 and 4×4 are represented by a single point. For each of pixel group types, experimental results using search approach of MRFME_1, MRFME_2 and MRFME_3 are shown in Table 27, Table 29 and Table 31. For each of the pixel group types, MRFME_1 has better results compared to the other two approaches in terms of maintaining the picture quality compared to the full search.

Seq.	SEA		MRFME_1		MRFME_2		MRFME_3	
	T(sec)	YPSNR	T(sec)	YPSNR	T(sec)	YPSNR	T(sec)	YPSNR
Bridge	512	39.32	83	39.31	71	39.32	70	39.34
Dance	149	45.8	20	45.78	13	45.73	13	45.73
Hall	268	37.18	36	37.18	39	37.07	37	37.07
CarPark	731	41.61	125	41.60	113	41.59	112	41.59
Street	1011	33.93	152	33.94	160	33.89	188	33.79

Table 27: 2×2 group with MRFME

Seq.	MRFME_1		MRFME_2		MRFME_3	
	SpeedUp	Δ YPSNR	SpeedUp	Δ YPSNR	SpeedUp	Δ YPSNR
Bridge	6.17	+0.01	7.21	0.00	7.31	-0.02
Dance	7.45	+0.02	11.46	+0.07	11.46	+0.07
Hall	6.88	0.00	6.87	+0.11	7.24	+0.11
CarPark	5.85	+0.01	6.47	+0.02	6.53	+0.02
Street	6.65	-0.01	6.32	+0.04	5.38	+0.14

Table 28: 2×2 group speedup and PSNR loss against SEA

The speed up factor and loss in the image quality results for fast MRFME approaches against SEA based MRFME are shown in Table 28, Table 30 and Table 32. Results show that MRFME_1 has the best results with non-zero pixels of difference frames in a group of 2x4 being represented with single location.

Seq.	SEA		MRFME_1		MRFME_2		MRFME_3	
	T(sec)	YPSNR	T(sec)	YPSNR	T(sec)	YPSNR	T(sec)	YPSNR
Bridge	512	39.32	63	39.32	64	39.28	63	39.32
Dance	149	45.8	10	45.78	11	45.71	11	45.71
Hall	268	37.18	32	37.20	31	37.02	32	37.02
CarPark	731	41.61	104	41.61	103	41.59	103	41.59
Street	1011	33.93	136	33.97	133	33.79	133	33.75

Table 29: 2×4 group with MRFME

Seq.	MRFME_1		MRFME_2		MRFME_3	
	SpeedUp	Δ YPSNR	SpeedUp	Δ YPSNR	SpeedUp	Δ YPSNR
Bridge	8.13	0.00	8.00	+0.04	8.13	0.00
Dance	14.9	+0.02	13.55	+0.09	13.55	+0.09
Hall	8.36	-0.02	8.38	+0.16	8.38	+0.16
CarPark	7.03	0.00	7.10	+0.02	7.10	+0.02
Street	7.43	-0.04	7.60	+0.14	7.60	+0.18

Table 30: 2×4 group speedup and PSNR loss against SEA

Seq.	SEA		MRFME_1		MRFME_2		MRFME_3	
	T(sec)	YPSNR	T(sec)	YPSNR	T(sec)	YPSNR	T(sec)	YPSNR
Bridge	512	39.32	68	39.30	68	39.30	76	39.38
Dance	149	45.8	13	45.74	13	45.70	20	45.71
Hall	268	37.18	36	37.20	34	37.02	41	37.03
CarPark	731	41.61	110	41.61	110	41.58	116	41.59
Street	1011	33.93	144	33.99	196	33.75	159	33.79

Table 31: 4×4 group with MRFME

Seq.	MRFME_1		MRFME_2		MRFME_3	
	SpeedUp	**Δ YPSNR**	**SpeedUp**	**Δ YPSNR**	**SpeedUp**	**Δ YPSNR**
Bridge	7.53	+0.02	7.53	+0.02	6.74	-0.06
Dance	11.46	+0.06	11.46	+0.10	7.45	+0.09
Hall	7.44	-0.02	7.88	+0.16	6.54	+0.15
CarPark	6.65	0.00	6.65	+0.03	6.30	+0.02
Street	7.02	-0.06	5.16	+0.18	6.36	+0.14

Table 32: 4×4 group speedup and PSNR loss against SEA

7.6 Conclusions

This Chapter describes a fast multiple reference frames based motion estimation technique. In the very first reference frame of each motion vector search, a successive elimination algorithm is used to find the best motion vector in the search window. For the remaining reference frames, the difference between two consecutive reference frames is taken to locate the pixels which are different in the current reference frame from the already searched reference frame. Searching locations having non-zero difference is equivalent to the full search because the locations with zero difference have already been tested in the previous reference frame. Experimental results show that for the best searching approach MRFME_1 for 2×4 grouping, the speed up factor is 14.9 with maximum loss of 0.02 dB.

Chapter 8

A Multi-Pattern Search Algorithm

8.1 Introduction

Motion is the main source of temporal variations in videos. The Motion Estimation (ME) is a process that estimates spatial displacements of the same pixels in neighbouring reference frames. In many video coding standards, significant improvement in bit-rate reduction is achieved through the application of motion estimation and Motion Compensation (MC) techniques. The process of ME divides frames into a group of pixels known as block. Block Matching Algorithms (BMAs) are used to find out the best-matched block from the reference frame within a fixed-size search window. Displacement of best-matched block from the reference block is described as Motion Vector (MV). The best-matched is usually evaluated through a cost function based on Block Distortion Measure (BDM) such as Mean Square Error (MSE), Mean Absolute Error (MAE) or Sum of Absolute Difference (SAD).

The full search (FS) BMA, which searches all the candidate blocks within the search window exhaustively, introduces high computational complexity. This high computational complexity imposes a big hurdle for real-time coding of videos. Under this motivation, many fast BMAs [58] - [69] based on heuristic patterns have been proposed to achieve fast motion estimation with a similar block distortion compared to FS and with a less computational complexity. These include four-step search (4SS) [60], diamond search (DS) [61], kite-cross diamond search (KCDS) [66], modified DS (MODS) [67], and cross-diamond-hexagonal search (CDHS) [68], etc. DS introduces a diamond shape searching pattern and unrestricted searching steps. KCDS utilizes cross-centre-biased MV distribution property. It employs a small cross-shaped search pattern in the first and second steps. MODS uses a dynamic threshold value to perform any time search stop. CDHS combines cross, diamond and hexagon shapes in the search pattern.

For fast motion estimation, a novel direction based Multi-Pattern search (MP search) algorithm has been proposed. It starts with a small cross in the first step. The second step identifies the trend of Motion Vector (MV) direction. Based on the direction of MV, obtained in second step, if the current minimum BDM point is along the horizontal or vertical axis of previous minimum BDM point then three new points forming a T shape with the previous minimum BDM point, are selected to perform a search in the next step. If the current minimum BDM point does not coincide with any axis of the previous minimum BDM then three new points, one along the direction of previous minimum BDM point and two along the direction of rectangular components of previous MV are checked for the minimum BDM. The proposed MP search algorithm is explained in detail in the following sections.

8.2 Multi-Pattern Search

The proposed MP search is based on the observation that when a motion vector points to a location other than the starting point during a search step then its direction tends to remain within +90 to -90 degrees in the following search step. This observation was testified on the sequences used in the experimental evaluation. For all of the sequences, the full search BMA was used and not a single MV violated above mentioned observation. Based on this observation, the number of candidate points

within a search window is reduced by avoiding the points which are in the region beyond a +90 and -90 degree of the motion vector. Further reduction in the remaining candidate points is achieved through the employment of different search patterns.

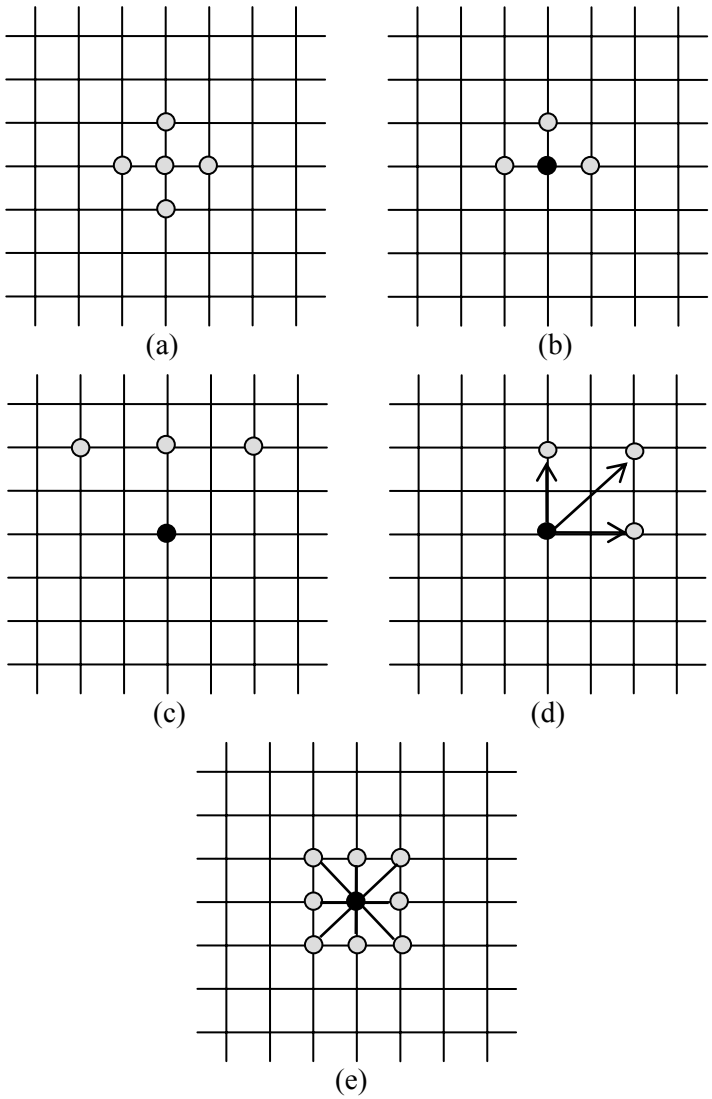


Figure 63: Search patterns for the MP search (a) small cross shaped pattern (SCSP) (b) small triangle shaped pattern (STSP) (c) large T-shaped pattern (LTSP) (d) large rectangular-directional search pattern (LRSP) (e) small double cross shaped pattern (SDCSP)

The MP search algorithm employs five different search patterns as illustrated in Figure 63. The black dot in these patterns represents the minimum BDM point of previous search step. The first pattern, a Small Cross Shaped Pattern (SCSP), consists

of 5 search points including the start point at the centre. A block search always starts with this pattern. This pattern helps to identify the direction of the flow of motion and helps early termination for static background based blocks. After the SCSP, Small Triangle Shaped Pattern (STSP) searches three new points surrounding previous optimal search point identified through the SCSP to further confirm the motion direction. The STSP shown in Figure 63 is assumed for an upward minimum BDM point from the previous step otherwise, this shape can be in any of four directions of the SCSP.

After the first two steps, further search is always performed using either Large T-shaped (LTSP) or Large Rectangular-directional Search Pattern (LRSP). When a minimum BDM is in the direction of previous minimum BDM then a T pattern is used which helps to avoid unnecessary checking of directionally unrelated points. When a minimum BDM point does not fall along the direction of the previous BDM then a rectangular direction based LRSP pattern is used. Thus, in a multi-pattern approach, checking points which have a very low probability to follow the motion inertia are avoided and hence complexity of ME is reduced. At any stage of the search, if previous minimum BDM is also the current minimum BDM then a Small Double Cross Shaped Pattern (SDCSP) is used to searches 8 new points surrounding minimum BDM. As seen in Figure 63, except for pattern (a) and (e), all the patterns perform testing on three new points. These three points based patterns reduce the complexity of motion vector search by avoiding unnecessary point checking. The multi-pattern search algorithm can be described in the following steps:

Step 1: A minimum BDM point is found by 5 checking points using SCSP. If the start point is the minimum BDM point then the search terminates here otherwise go to step 2.

Step 2: Minimum BDM point of step 1 is surrounded in a small triangle shaped pattern to search 3 new points for a minimum BDM. Then go to step 3.

Step 3: Considering the minimum BDM point of the previous step, a large T-shaped search pattern is formed. If a previous minimum BDM point is also a minimum BDM point of this step or minimum BDM point of step touches the boundary of the search window then go to step 5. If a minimum BDM point found in this step is along the axis of previous minimum BDM point then repeat step 3 for a newly found minimum BDM point otherwise go to step 4.

Step 4: Considering the minimum BDM point of the previous step, a large rectangular-directional search pattern is employed. If the previous minimum BDM point is also the minimum BDM point of this step or the minimum BDM point touches the boundary of search window then go to step 5. If the minimum BDM point found is along any of rectangular directions of previous minimum BDM point then go to step 3 otherwise repeat step 4.

Step 5: Use the small double cross search pattern to check 8 new search points. The minimum BDM point found in this step is the best matched point for motion vector representation. Terminate search.

To illustrate the workflow of the MP search, two different examples of motion vector search are shown in Figure 64 with different search paths with a search range of 7.

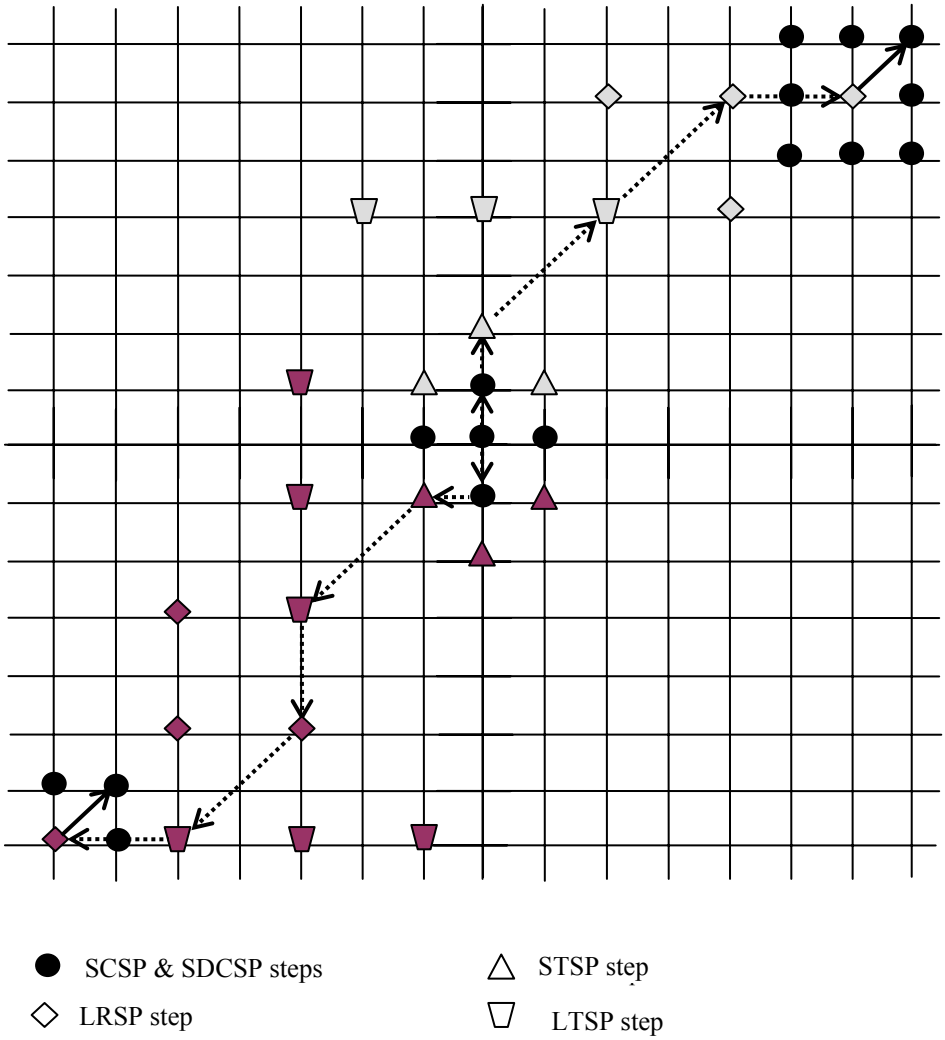


Figure 64: Two different search paths of MP search

8.3 Simulation Results

Performance of the proposed MP search has been evaluated using five different image sequences shown in Table 33. These sequences can be divided into three groups: Akiyo and Carphone with a low motion activity, Foreman and Tennis with normal motion activity and Football with high motion activity.

Sequences	Frame rate	Frame size	Frames
Akiyo	30	352x288	300
Carphone	30	352x288	350
Football	30	352x288	260
Foreman	30	352x288	300
Tennis	30	352x288	300

Table 33: Image sequences used in experiment

While performing the experiment, the Sum of Absolute Difference (SAD) is used as a Block Distortion Measure (BDM). The block size is 16×16 and the search range is 15 with GOP size 64. Different criteria are used for evaluating the proposed MP search algorithm. They are the Mean Square Error (MSE), average number of search points per motion vector, relative complexity reduction (RCR), and SpeedUp ratio. Table 34 shows MSE results for the comparison of different BMAs in terms of reconstructed image quality. Results show that the proposed MP search has always performed better than other BMAs except for the Tennis sequence with very similar MSE results.

BMA	Akiyo	Carphone	Football	Foreman	Tennis
FS	1.82	38.47	206.44	33.35	31.51
DS	1.84	46.73	284.02	66.80	63.58
CDH	1.89	47.56	288.27	70.42	64.42
MP	1.84	46.97	286.36	69.54	65.13

Table 34: Average MSE results

Table 35 shows results of average number of search points per MV. Note that considering the search window size; this value is constant for FS. Results show that

the proposed MP search has the lowest average number of search points per MV for each type of image sequence.

BMA	Akiyo	Carphone	Football	Foreman	Tennis
FS	961	961	961	961	961
DS	16.04	42.80	94.01	74.63	35.58
CDH	8.98	34.19	65.90	55.24	24.54
MP	6.83	17.45	30.01	25.47	13.03

Table 35: Average number of search points per MV

Table 36 and Table 37 show computational power saving for each sequence in terms of Relative Complexity Reduction (RCR) and SpeedUp ratio. These are calculated using the following relationships:

$$RCR = (1 - NS_{MP} / NS) \times 100\ \% \quad (8.1)$$

$$SpeedUp = NS / NS_{MP} \quad (8.2)$$

where NS_{MP} and NS represents total number of search points using the proposed MP search and other Fast search approaches, respectively. These two parameters, RCR and SpeedUp ratio, describe how much computational power is saved using the proposed approach in comparison to the DS and CDH search approach.

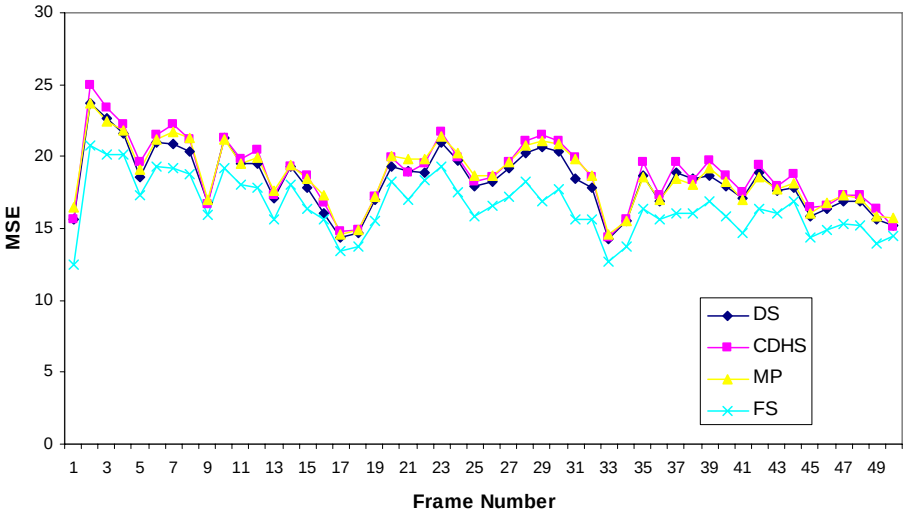
BMA	Akiyo	Carphone	Football	Foreman	Tennis
DS	2.06	1.91	2.67	2.42	2.31
CDH	1.15	1.53	1.87	1.79	1.60

Table 36: MP Speedup factor against DS and CDH

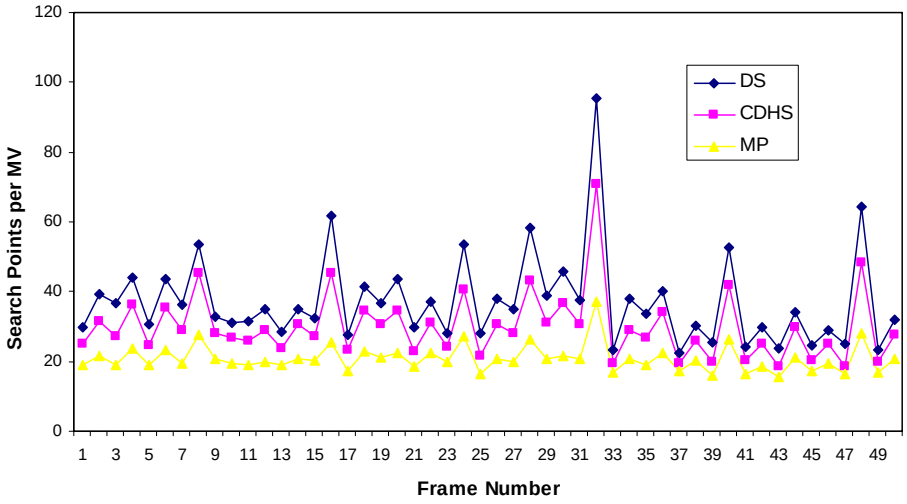
BMA	Akiyo	Carphone	Football	Foreman	Tennis
DS	51.46	47.64	62.55	58.68	56.71
CDH	13.04	34.64	46.52	44.13	37.50

Table 37: RCR (%) for MP compared to DS and CDH

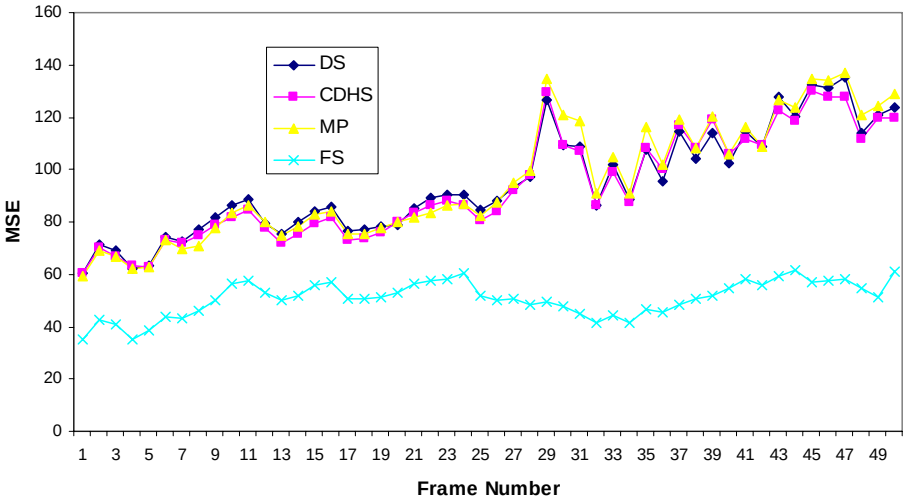
Experimental results of Table 36 and Table 37 show that the proposed approach can reduce the computational complexity significantly with very little change in MSE.



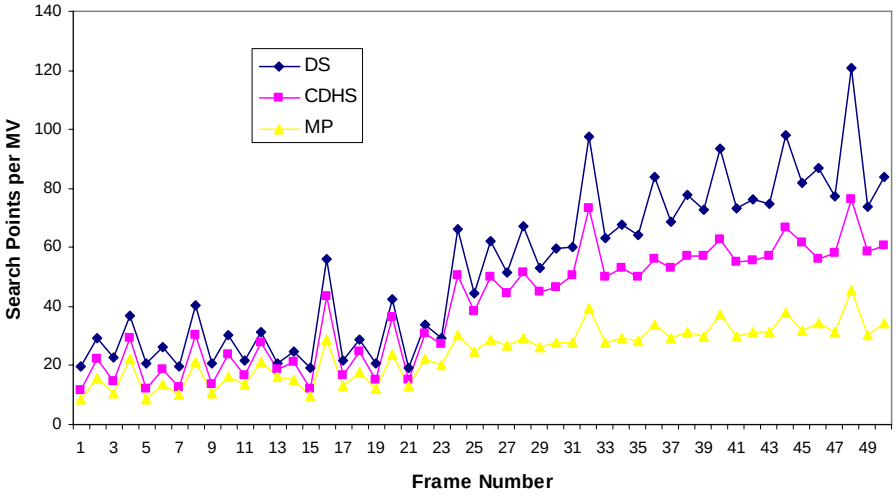
(a) Carphone



(b) Carphone



(c) Tennis



(d) Tennis

Figure 65: Frame-by-frame comparison (a) MSE results for Carphone (b) Search points per MV for Carphone (c) MSE results for Tennis (d) Search points per MV for Tennis

Figure 65 illustrate frame-by-frame performance comparison of the proposed approach against the FS, DS, CDH search approaches for the 'Carphone' and 'Tennis' videos.

8.4 Conclusions

In this chapter, a novel multi-pattern search technique has been presented to perform fast motion estimation. The proposed search pattern starts with small cross shaped

and small triangular shaped patterns. Afterwards based on the previous optimal search point, it selects either a T-shaped pattern or rectangular component-directional pattern. A high reduction in computational complexity is achieved by using the proposed technique. Performance of the implemented MP block search is compared against DS and CDHS BMAs. MSE and average number of search points per motion vector results obtained through an experimental evaluation show that the processing speed can be improved significantly by using the proposed approach while maintaining a comparable image quality.

Chapter 9

Conclusions

The research presented in this thesis focused on diverse coding techniques specific to surveillance videos. The foremost aim of this work was to propose such approaches to improve the storage capacity and bandwidth utilisation with less computational complexity. The proposed coding techniques to improve compression and processing efficiency have been described to improve on the conventional techniques. These techniques are developed under the motives laid down in Chapter 1. Most of the work has been tested and evaluated on a set of typical surveillance videos.

9.1 Conclusions

In the previous chapters, the basic techniques for video coding in the state-of-the art video coders have been presented. The object-based coding approach in MPEG-4 has been explained. After the background study of state-of-the art approaches, the achievements in developing the surveillance centric coding techniques can be summarised as below.

1. A technique to improve on the compression efficiency for surveillance videos has been presented. The architecture of the scalable video coding has been modified to become surveillance centric coding. The modified architecture offers a better byte saving performance. The architectural modifications in the SVC help to deal each GOP of the video with different coding parameters. This approach shows that the application of scalable video coding with an event driven approach improves the transmission and storage efficiency.

2. After introducing the SCC architecture, a novel approach to implementing foreground based SCC has been proposed. The foreground pixels are selected by using the bounding boxes of the VCA modules. This approach has the benefit of being free from shape coding and background coding as compared to MPEG-4 object based coding. In addition to this, scalable video codec can be used to exploit the scalability features as described in the SCC.

3. Different experimental results showed that the motion compensated temporal filter (MCTF) with higher levels of filtering helps to remove the temporal redundancies present in the surveillance videos. This approach has better RD-performance than the block-based coding approaches, H.264.

4. A search technique with higher processing efficiency with a visual quality equivalent to the full search approach and selective search strategy specific to surveillance videos is presented. Two approaches to perform selective motion estimation, GOP based and Frame based, are described where Frame based approach performed better. The visual quality for both the approaches is the same as that of the full search.

5. To improve the selective motion estimation further, a selective block search technique has been proposed. The selection of the block is based on the novel approach where a motion detection module is used to provide the location of candidate blocks. Although, this approach performs a fewer number of block matching steps yet the overall processing efficiency of the system is close to a

Frame based selective search. This is because of the overhead complexity added by the bounding box matching algorithm to locate the candidate block in the SCC.

6. After introducing selective motion estimation approaches, another novel approach, tracker-based motion estimation has been proposed where surveillance video object motion tracker information is used to calculate the motion vectors. A unique motion track is calculated for each object of the surveillance video. The distance representing the displacement of the object between the current and the reference frame is taken as a value to calculate motion vectors after identifying and matching the track in the two frames. This approach performs a faster calculation of the motion vectors but it degrades the visual quality of the video depending on the nature of the movement represented by the foreground object.

7. Finally, under the same motivation of achieving processing efficiency without loss in visual quality, fast full search approaches are explored. A fast search approach for multiple reference frames specific to surveillance videos has been proposed. This approach is based on considering different points between the two consecutive reference frames and then using these different points to avoid unnecessary block matching steps. This search approach is specific to surveillance videos with motion estimation based on multiple reference frames.

8. To improve the processing efficiency of the SCC, a multi-pattern search approach is proposed. This approach improves the processing efficiency with some loss in visual quality compared to the full search technique. However, it maintains comparable visual quality with respect to other fast search techniques for example the diamond search.

The current achievements described above show the performance of the SCC in terms of compression efficiency and processing efficiency.

9.2 Key Contributions

With the motivation to achieve the objectives laid down in Chapter 1, several contributions have been made. These contributions are classified into two groups: Efficient Compression and Efficient Motion Estimation.

9.2.1 Efficient Compression

After distinguishing the potential benefits of using the Scalable Video Coding (SVC) techniques for surveillance videos, the SVC framework was adopted to implement the Surveillance Centric Coding (SCC) paradigm introduced in Chapter 5. One of the drawbacks in using the SVC framework for the SCC was the sliding filtering window between the consecutive GOPs. This sliding window was acting like a bond between the two GOPs, restraining the possibility of treating each GOP independently. So, the architectural modifications proposed and presented in Chapter 5 helped to break up this inter-GOP bondage. After the removal of the GOP dependency, the second step accomplished was to achieve the ability to deal with each GOP according to the SCC requirements while still maintaining the SVC properties. So, the communication link between the Video Content Analysis (VCA) module and a single GOP was established. The analysis of each GOP generated by the VCA under the requirements of the SCC was used to exploit the SVC properties of each GOP. Thus, the bit-stream generated by the SCC consisted of GOPs with different scalability features to compress the surveillance videos with higher compression efficiency while it not compromising the important information from the surveillance standpoint.

> So, a flexible framework of the SCC was developed after proposing the architectural modifications for the SVC. Higher compression efficiency was achieved by establishing the link between the SCC and the VCA.

After the implementation of the SCC paradigm, another novel approach was proposed to improve the compression efficiency. In this approach, foreground objects are detected by the VCA by forming rectangular windows around objects. The first frame of the sequence is used as background and the rest of the frames contain only the foreground pixels while the background pixels are set to zeros. This showed efficient

compression but due to the use of block based coding approaches, lack of sharpness in the foreground boundary was observed. The major advantage of using this approach is its implementation in the SCC framework. Thus, in addition to avoiding shape coding and other object based coding techniques, the scalabilities features are inherited through the SCC framework offering the potential of improving the compression efficiency by exploiting the scalability features in each GOP.

So, a novel approach focusing on the foreground pixels was proposed and implemented in the framework of the SCC to achieve a better compression efficiency.

9.2.2 Efficient Motion Estimation

Apart from storage issues for surveillance videos, the main challenge was to compress the surveillance videos as quickly as possible. As the Motion Estimation (ME) is the most processing intensive part of a codec; therefore, efficient techniques to perform fast motion estimation were explored. In addition to proposing some efficient motion estimation algorithms, the task of using the SCC framework described in the last section in terms of integrating proposed motion estimation algorithms was accomplished.

A novel approach for performing selective motion estimation was proposed whereby the object detection information generated by the VCA was used to flag the frames which did not contain any moving object. Based on this analysis, different selective motion estimation approaches were proposed which included: GOP level selective motion estimation, Frame level selective motion estimation and block level motion estimation.

In the GOP level selective motion estimation, the decision to perform or skip the motion estimation was enforced at the GOP level. So, the only way of skipping the motion estimation for a particular GOP was the scenario where there was not a single frame of the GOP identified as containing a moving object. Due to the very low probability of such a scenario, this scheme contributed a little improvement even for moderately busy locations. The second drawback of the GOP level selective motion estimation was its dependency on the GOP size. So, with smaller GOP size, there was

a higher probability of such GOPs occurring which do not have any frame detected containing a moving object. To counter the issues faced in the GOP level selective ME, a Frame level selective ME was proposed where the decision to perform or skip the ME was taken for each frame independently of other frames. Once again, this approach was integrated into the SCC framework. Evidently, the Frame level selective ME performed better than the GOP level. The key observation for the Frame level ME was the imposing of the ME decision for all the blocks of the frame irrespective of the location and size of the moving objects in the frame.

> Novel approaches to performing fast motion estimation were proposed. By using the VCA, selective motion estimation helped to speed up the compression process by avoiding unnecessary computations.

With the motivation of evolving selective ME from Frame to macroblock level, a block level selective ME was proposed. Two major challenges to implementing this approach were (i) identification of the macroblock as part of a moving object and (ii) locating those blocks while performing ME. As explained in Chapter 5, the foreground pixels identified by the VCA were isolated from the background pixels by using the information of the rectangular bounding boxes. A similar approach was used to implement the Block level selective motion estimation.

> A novel approach for implementing the block level selective motion estimation was proposed.

After the paradigm of selective motion estimation, a novel way of performing efficient motion estimation through reusing the information of a surveillance video object tracker was proposed. In this approach, a real-time object tracker was used which generates information for each unique object with a unique track identity. In addition to this, objects were bounded in a rectangular box. So, instead of performing any kind of motion estimation for any block of the surveillance video, the motion vectors are calculated through the information generated by the object tracker. This approach had a drawback of miscalculating some of the motion vectors corresponding to the same object; ultimately, reducing the visual quality.

Another novel approach for fast motion estimation was proposed where instead of using motion estimation module of the SCC, the surveillance video tracker provided the information to calculate the motion vectors.

Multiple reference frame based motion estimation increases the computational complexity with every extra reference frame. In order to address this problem, a fast full search for multiple reference frames based ME was proposed. This scheme was based on the Successive Elimination Algorithm (SEA): a fast full search approach. This approach reduced the processing power for surveillance videos. One drawback was the extra memory used.

Fast multi-frame motion estimation for surveillance videos was proposed in order to maintain visual quality.

Finally, another fast ME search algorithm, multi-pattern search algorithm, was proposed to find approximate calculations as in the case of the Diamond search. This algorithm is valid for any kind of videos, surveillance or non-surveillance.

A fast motion estimation search algorithm was proposed. Computational complexity is reduced through acceptable compromise on visual quality.

9.3 List of Publications

The work presented in this thesis has lead to the following research publications. The author of the thesis is the main contributor to all the stages of the publication work from concept development to submission.

1. M. Akram and E. Izquierdo, "Fast multi-frame motion estimation for surveillance videos," *in the Proc. of IEEE 17th International Conference on Image Processing (ICIP 2010),* pp. 753-756, September 2010.
2. M. Akram and E. Izquierdo, "Selective block search for surveillance centric motion estimation," *in the Proc. of IEEE 52nd International Symposium ELMAR 2010,* pp. 93-96, September 2010.

3. M. Akram and E. Izquierdo, "A multi-pattern search algorithm for block motion estimation," *in the Proc. of IEEE 12th International Asia-Pacific Web Conference (APWEB 2010),* pp. 407-410, April 2010.
4. M. Akram, N. Ramzan, and E. Izquierdo, "Selective motion estimation for surveillance videos," *in the Proc. of International Conference on User Centric Media (UCMEDIA 2009),* pp. 199-206, Dec 2009.
5. M. Akram, N. Ramzan, and E. Izquierdo, "Efficient motion estimation for video coding in wireless surveillance applications," *in the Proc. of IEEE International Conference on Ultra Modern Telecommunications (ICUMT 2009),* Oct 2009.
6. M. Akram, N. Ramzan, and E. Izquierdo, "Event based video coding architecture," *in the Proc. of IET 5th International Conference on Visual Engineering (VIE 08),* pp. 807-812, July 2008.
7. T. Zgaljic, N. Ramzan, M. Akram, and E. Izquierdo, "Surveillance centric coding," *in the Proc. of IET 5th International Conference on Visual Engineering (VIE 08)*, pp. 835-839, July 2008.

References

[1] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in Information and Decision Processes, R.E. Machol, Ed. New York: McGraw-Hill, 1960, pp. 93-126.

[2] S. Wan, "Rate-Distortion optimization for motion-compensated prediction in video coding," *PhD thesis, Department of Electronic Engineering, Queen Mary, University of London*, 2007.

[3] R. Gallager, Information theory and reliable communication. New York: JohnWiley and Sons, 1968.

[4] D. Slepian, Ed., Key papers in the development of information theory. New York: IEEE Press, 1974.

[5] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379-423 & 623-656, Jul. & Oct. 1948.

[6] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23-50, Nov. 1998.

[7] M. Brotherton, H. Quan, D. Hands, and K. Brunnstrom, "Subjective multimedia quality assessment," *IEICE Transactions on Fundamentals*, vol. E89-A, no. 11, pp. 2920-2932, Nov. 2006.

[8] VQEG Report, "Final report from the video quality experts group on the validation of objective models of video quality," Jun. 2000.

[9] VQEG Report, "Draft final report from the video quality experts group on the validation of objective models of video quality assessment," Aug. 2003.

[10] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations Research*, vol. 11, no. 3, pp. 399- 417, 1963.

[11] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 9, pp. 1445-1453, Sept. 1988.

[12] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, vol. 2, no. 2, 160-175, Apr. 1993.

[13] G. M. Schuster and A. K Katsaggelos, "An optimal quadtree-based motion estimation and motion-compensated interpolation scheme for video compression," *IEEE Transactions on Image Processing*, vol. 7, no. 6, pp. 909- 912, Nov. 1998.

[14] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74-90, Nov. 1998.

[15] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Transactions on Circuits Systems for Video Technology*, vol. 7, no. 1, pp, 246-250, Feb. 1997.

[16] A. Vetro, T. Haga, K. Sumi, H. Sun, "Object-based coding for long-term archive of surveillance video," *Technical Report*, TR-2003-98, MERL, 2003.

[17] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *Pro. Computer Vision Pattern Recognition*, Fort Colins, CO, June 1999.

[18] A. Elgammal, D. Harwood, and L. S. Davis, "Nonparametric background model for background subtraction," *Pro. 6th European Conf. Computer Vision*, 2000.

[19] T. Matsuyama, T. Ohya and H. Habe, "Background subtraction for non-stationary scenes," *Proc. 4th Asian Conf. on Computer Vision*, pp.662-667, 2000.

[20] Y. Yu and D. Doermann, "Model of object-based coding for surveillance video," *Proc. IEEE International Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 693-696, 2005.

[21] T. Nishi and H. Fujiyoshi, "Object-based video coding using pixel state analysis," *Proc. of IEEE International Conf. on Pattern Recognition (ICPR)*, 2004.

[22] A. Cavallaro, O. Steiger and T. Ebrahimi, "Perceptual prefiltering for video coding," Proc. *IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing (ISIMP)*, 2004.

[23] J. Su, Q. Liu and T. Ikenaga, "Motion detection based motion estimation algorithm for video surveillance application," *Proc. IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, pp. 481-484, Dec 2009.

[24] ITU-T Rec. H.264/ISO/IEC 11496-10, "Advance Video Coding," *Final Committee Draft, Document JVT-E022*, September 2002.

[25] ITU-T Rec. H.264/ISO/IEC 11496-10, "Advance Video Coding," *Final Committee Draft, Document JVT-g050*, March 2003.

[26] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560-576, July 2003.

[27] I. E. G. Richardson, "Tutorial on H.264/AVC Video Compression Standard," April 2003. [Online]. Available: http://www.vcodex.com/h264.html [Accessed: November 19, 2010].

[28] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC advance video coding standard: overview and introduction to the fidelity range extension," *in the Proc. of the SPIE Conference on Applications of Digital Image Processing XXVII*, August 2004.

[29] T. Schierl, T. Wiegand, "Mobile video transmission using SVC", *IEEE Trans. Circuits Syst. Video Technology*, vol. 17, no. 9, September 2007.

[30] M. Wien, R. Cazoulat, A. Graffunder, A. Hutter, P. Amon, "Real-time system for adaptive video streaming based on SVC", *IEEE Trans. Circuits Syst. Video Technology,* vol. 17, no. 9, pp. 1227-1237, September 2007.

[31] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 17, no. 9, pp. 1103-1120, September 2007.

[32] F. Ziliani, "The importance of 'Scalability' in video surveillance architectures", *IEE International Symposium on Imaging for Crime Detection and Prevention*, June 2005.

[33] May, J. The, P. Hobson, F. Ziliani, J. Reichel, "Scalable video requirements for surveillance applications", *IEE Intelligent Distributed Surveillance System*, pp. 17–20, Feb. 2004.

[34] M. Mrak, N. Sprljan, T. Zgaljic, N. Ramzan, S. Wan and E. Izquierdo, "Performance evidence of software proposal for Wavelet Video Coding *Exploration group", Technical Report ISO/IEC JTC1/SC29/WG11/MPEG2006/M13146,* (2006).

[35] Y. Andreopoulos, M. Van der Schaar, A. Munteanu, J. Barbarien, P. Schelkens, J. Cornelis, "Complete-to-overcomplete discrete wavelet transforms for scalable video coding with MCTF", *Proc. Visual Communication and Image Processing (VCIP 2003)*, pp. 719–731, July 2003.

[36] J.-R. Ohm, "Three-dimensional Subband Coding with Motion Compensation," *IEEE Trans. Image Processing*, vol. 3, pp. 559–571, September 1994.

[37] W. Sweldens and P. Schroder, "Building your own wavelets at home," *Wavelets in Computer Graphics*, pp. 15–87, ACM SIGGRAPH Course notes, 1996.

[38] M. Mark and E. Izquierdo, "Spatially adaptive wavelet transform for video coding with multi-scale motion compensation" *Proc. IEEE International Conference on Image Processing*, vol. 2, pp. 317–3320, Sep. 2007.

[39] T. Zgaljic, N. Sprljan and E. Izquierdo, "Bitstream syntax description based adaptation of scalable video", *Integration of Knowledge, Semantics and Digital Media Technology (EWIMT 2005)*, pp. 173–176, 30 Nov. 2005.

[40] M. Mrak, "Motion scalability for video coding with flexible spatio-temporal decompositions", *PhD thesis, Department of Electronic Engineering, Queen Mary, University of London*, 2007.

[41] T. Zgaljic, N. Sprljan and E. Izquierdo, "Bit-stream allocation methods for wavelet based scalable video coding", *Proc. 2nd International Mobile Multimedia Communications Conference (Mobimedia 2006)*, September 2006.

[42] S.-J. Choi, J.W. Woods, "Motion-compensated 3D subband coding of vidoe", *IEEE Transaction on Image Processing*, vol. 8, pp. 155–167, April 1996.

[43] V. Bottreau, M. Benetiere, B. Felts, B. Pesquet-Popescu, "A fully scalable 3D subband video codec", *Proc. of IEEE Conference of Image Processing*, vol.2, pp. 1017–1020, October 2001.

[44] A. Secker, D. Taubman, "Lifting-based invertible motion adaptive transform (LIMAT) framework for highly scalable video compression", *IEEE Transaction on Image Processing*, vol. 12, pp. 1530–1542, December 2003.

[45] Y. Andreopoulos, A. Munteanu, J. Barbarien, M. van der Shaar, J. Cornelis and P. Schelkens, "In-band motion compensated temporal filtering", *Signal Processing: Image Communication*, vol. 19, pp. 653–673, August 2004.

[46] R. Xiong, J. Xu, F. Wu, S. Li, "In-scale motion aligned temporal filtering", *Proc. IEEE International Symposium on Circuits and Systems*, May 2006.

[47] R. Xiong, J. Xu, F. Wu, S. Li, "Studies on spatial scalable frameworks for motion aligned 3D wavelet video coding", *Proc. Visual Communications and Image Processing (VCIP 2005)*, July 2005.

[48] M. Wien, "Variable block-size transforms for H.264/AVC", *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, pp. 604–613, July 2003.

[49] W. B. Pennebaker, J. L. Mitchell, "JPEG still image data compression standard", Van Nostrand Reinhold, New York, 1993.

[50] D. Taubman, M. W. Marcellin, "JPEG2000 image compression: fundamentals, standards and practice", Kluwer Academic Publishers, 2002.

[51] JPEG 2000 image coding system, *ISO/IEC 15 444*.

[52] N. Sprljan, M. Mrak, G. C. K. Abhayaratne, E. Izquierdo, "A scalable coding framework for efficient video adaptation", *Proc. 6th International Workshop*

on Image Analysis for Multimedia Interactive Services (WIAMIS 2005), April 2005.

[53] S. –T. Hsiang, J. W. Woods, "Embedded video coding using invertible motion compensated 3D subband/wavelet filter bank", *Signal Processing: Image Communication*, vol.16, Iss. 8, pp. 705–724, May 2001.

[54] T. Zgaljic, N. Sprljan, E. Izquierdo, "Scalable video adaptation based on bitstream syntax description", *Proc. Workshop on Immersive Communication and Broadcast Systems (ICOB 2005)*, October 2005.

[55] MPEG-4 Requirements version 17, ISO/IEC JTC1/SC29/WG11 N4319, Sydney, July 2001.

[56] E. S. Jang, "Low-Complexity MPEG-4 shape encoding towards realtime Object-Based applications", *ETRI Journal*, vol. 26, no. 2, April 2004.

[57] T. Ebrahimi, C. Horne, "MPEG-4 natural video coding – an overview", *Signal Processing: Image Communication*, vol. 15, no. 4–5, pp. 365–385, 2000.

[58] M. Ghanbari, "The cross search algorithm for motion estimation," *IEEE Trans. Communication.*, vol. 38, no.7, pp. 950 – 953, July 1990.

[59] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 4, pp. 438 – 442, Aug. 1994.

[60] L. M. Po and W. C. Ma, "A novel four step search algorithm for fast block motion estimation," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 6, pp. 313 – 317, June 1996.

[61] S. Zhu and K. K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, pp. 287 – 290, Feb. 2000.

[62] O. T.-C. Chen, "Motion estimation using a one directional gradient descent search," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 10, pp. 608 – 616, June 2000.

[63] Y. Nie and K-K. Ma, "Adaptive rood pattern search for fast block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 11, pp. 1442 – 1449, Dec. 2002.

[64] C. Zhu, X. Lin and L-P. Chau, "Hexagon based search pattern for fast block motion estimation," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 12, pp. 349 – 355, May 2002.

[65] C-H. Cheung and L-M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 12, pp. 1168 – 1177, December 2002.

[66] L-M. Po and C-H. Cheung, "A novel kite-cross-diamond search algorithm for fast block matching motion estimation," *IEEE ISCAS*, vol. 3, pp. 729 – 732, May 2004.

[67] X. Yi and N. Ling, "Rapid block-matching motion estimation using modified diamond search," *IEEE ISCAS*, vol. 6, pp. 5489 – 5492, May 2005.

[68] C-H. Cheung and L-M. Po, "Novel cross-diamond-hexagonal search algorithm for fast block motion estimation," *IEEE Trans. Circuit Syst. Video Technol.*, vol. 7, no.1, pp. 16 – 22, Feb. 2005.

[69] Y. Nie and K. K. Ma, "Adaptive rood pattern search for fast block matching motion estimation," *IEEE Trans. Image Process.*, vol. 11, no.12, pp. 1442 – 1449, Dec. 2002.

[70] C. Stauffer, and W.E.L. Grimson, "Learning patterns of activity using real time tracking," *IEEE Trans. on Pattern Analysis and Mach. Intel.,* vol. 22, pp. 747-757, 2000.

[71] M. Mrak, N. Sprljan, T. Zgaljic, N. Ramzan, S. Wan and E. Izquierdo, "Performance evidence of software proposal for wavelet video coding exploration group," *Technical Report ISO/IEC JTC1/SC29/WG11/MPEG2006/M13146,* (2006)

[72] T. Zgaljic, N. Ramzan, M. Akram, E. Izquierdo, R. Caballero, A. Finn, H. Wang and Z. Xiong, "Surveillance centric coding," *In 5th International Conference on Visual Information Engineering (VIE 2008)*, pp. 835-839, July 2008.

[73] M. Akram, N. Ramzan and E. Izquierdo, "Event based video coding architecture," *In 5th International Conference on Visual Information Engineering (VIE 2008)*, pp. 807-812 July 2008.

[74] T. Zgaljic, N. Sprljan and E. Izquierdo, "Bit-stream allocation methods for scalable video coding supporting wireless communications," *Signal Processing: Image Communications*, vol. 22, pp. 298-316, March 2007.

[75] Recommendation ITU-T BT 500.10: Methodology for the Subjective Assessment of the Quality of Televisions Pictures, (2000).

[76] W. Li and E. Salari, "Successive elimination algorithm for motion estimation," *IEEE Trans. Image Processing*, vol. 4, pp. 105—107, Jan. 1995.

[77] H-S. Wang and R. M. Mersereau, "Fast algorithm for the estimation of motion vectors," *IEEE Trans. Image Processing*, vol. 8, pp. 435—438, Mar. 1999.

[78] X. Q. Gao, C. J. Duanmu and C. R. Zou, "A multilevel successive elimination algorithm for block matching motion estimation," *IEEE Trans. Image Processing*, vol. 9, pp. 501—504, Mar. 2000.

[79] S-M. Jung, S-C. Shin, H. Baik and M-S. Park, "New fast successive elimination algorithm," *In proc. 43rd IEEE Midwest Symp. on Cct. and Sys.,* pp. 616—619, Aug. 2000.

[80] S-M. Jung, S-C. Shin, H. Baik and M-S. Park, "Nobel successive elimination algorithm for the estimation of motion vectors," *In proc. 43rd IEEE Microelectronic System Education,* pp. 332—335, 2000.

[81] J-N. Kim, S-C. Byun, Y-H. Kim and B-H. Ahn, "Fast full search motion estimation algorithm using early detection of impossible candidate vectors," *IEEE Trans. Signal Processing*, vol. 50, pp. 2355—2365, Sept. 2002.

[82] S-M. Jung, H. Baik, S-H. Lee and M-S. Park, "Advanced multilevel successive elimination algorithms using the locality of pixels in block," *Journal of Korean Society for Industrial and Applied Mathematics-IT seris,* vol. 8, pp. 75—86, 2004.

[83] Y. Song, Z. Liu, T. Ikenaga, and S. Goto, "Enhanced strictly multilevel successive elimination algorithm for fast motion estimation," *IEEE Symp. on Cct. and Sys.,* pp. 3659—3662, May 2007.

[84] ITU-T Rec. H.264/ISO/IEC 11496-10, "Advance Video Coding," *Final Committee Draft, Document JVT-E022*, September 2002.

[85] ITU-T Rec. H.264/ISO/IEC 11496-10, "Advance Video Coding," *Final Committee Draft, Document JVT-g050*, March 2003.

[86] H. F. Ates and Y. Altunbasak, “SAD reuse in hierarchical motion estimation for the H.264 encoder,” *In proc. IEEE Acoustics, Speech and Signal Processing (ICASSP),* pp. 905—908, 2005.

[87] Y. Su and M.-T. Sun, “Fast multiple reference frame motion estimation for H.264/AVC,” *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 16, pp. 447—452, Mar. 2006

[88] M.-J. Chen, G.-L. Li, Y.-Y. Chiang and C-T. Hsu, “Fast multiframe motion estimation algorithm by motion vector composition for the MPEG-4/AVC/H.264 standard,” *IEEE Trans. Multimedia*, vol. 8, pp. 478—487, June. 2006

[89] S.-C. Hsia and Y.-C. Hung, "Fast multi-frame motion estimation for h264/avc system," *Journal of Signal, Image and Video Processing (SIViP)*, Feb. 2009.

[90] A. Pietrowcew, A. Buchowicz, W. Skarbek, "Bit-rate control algorithm based on local image complexity for video coding with ROI," *IEEE Conference on Advanced Video and Signal Based Surveillance*, pp.582-587, 2005.

[91] S. V. Leuven1, K. V. Schevensteen, T. Dams, and P. Schelkens, “An Implementation of multiple region-of-interest models in H.264/AVC,” *6th International Conference on Signal-Image Technology and Internet-Based Systems*, pp. 502 – 511, 2006.

[92] Y. Liu, Z. G. Li, Y. C. Soh and M. H. Loke, “Conversational video communication of H.264/AVC with region-of-interest concern,” *IEEE Conference on Image Processing (ICIP)*, pp. 3129 – 3132, Oct. 2006.

[93] P. Lambert, W. De Neve, Y. Dhondt, R. V. De Walle, “Flexible macroblock ordering in H.264/AVC,” *Journal of Visual Communication and Image Representation*, Volume 17, Issue 2, Introduction: Special Issue on emerging H.264/AVC video coding standard, April 2006, Pages 358-375, 2005.

[94] P. Baccichet, X. Zhu, and B. Girod, "Network-Aware H.264/AVC Region-of-Interest Coding for a Multi-Camera Wireless Surveillance Network", *Proc. Picture Coding Symposium, (PCS-06)*, Beijing, China, April 2006.

[95] P. Sivanantharasa, W.A.C. Fernando, H. Arachchi and Kodikara, “Region of interest video coding with flexible macroblock ordering,” *International Conference on Industrial and Information Systems*, pp. 596 – 599, Aug. 2006.

[96] Y. Zheng, J. Feng, H. Ma, and Y. Chen, "H.264 ROI coding based on visual perception," *5th International Conference on Visual Information Engineering (VIE)*, pp. 829 – 834, Aug. 2008.

[97] Y. Shi, S. Yue, B. Yin and Y. Huo, "A novel ROI based rate control scheme for H.264," *9TH International Conference for Young Computer Scientist*, pp. 77 – 81, 2008.

[98] Y. Liu, Z. G. Li and Y. C. Soh, "Region-of-interest based resource allocation for conversational video communication of H.264/AVC," *IEEE Trans. on Cct. and Sys. for Video Tech.*, vol. 18, pp. 134 – 139, Jan. 2008.

[99] P. Lambert and R. V. de Walle, "Real-time interactive region of interest in H.264/AVC," *Journal of Real-Time Image Processing*, vol. 4, pp. 67 – 77, March 2009

[100] M. Wang, T. Zhang, C. Liu and S. Goto, "Region-of-interest based dynamical parameter allocation for H.264/AVC encoder," *27th Conference of Picture Coding Symposium (PCS)*, pp. 93 – 96, 2009.

[101] T. Nishi, H. Fujiyoshi, "Object-based video coding using pixel state analysis," *IEEE International Conference on Pattern Recognition,* vol. 3, pp. 306-309, 2004.

[102] A. Hakeem, K. Shafique, M. Shah, "An object-based video coding framework for video sequences obtained from static cameras," *ACM International Conference on Multimedia,* pp. 608-617, 2005.

[103] T. Zgaljic, N. Ramzan, M. Akram, and E. Izquierdo, "Surveillance centric coding," *in the Proc. of IET 5th International Conference on Visual Engineering (VIE 08)*, pp. 835-839, July 2008.