Fast and Efficient Foveated Video Compression Schemes for H.264/AVC Platform

Deepak Singh

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela (India)

Fast and Efficient Foveated Video Compression Schemes for H.264/AVC Platform

Dissertation submitted in partial fulfillment

of the requirements of the degree of

Doctor of Philosophy

in

Electronics and Communication Engineering

by

Deepak Singh

(Roll Number: 511EC105)

based on research carried out

under the supervision of

Prof. Sukadev Meher

January, 2017

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela (India)

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela (India)

January 10, 2017

Certificate of Examination

Roll Number: *511EC105*
Name: *Deepak Singh*
Title of Dissertation: *Fast and Efficient Foveated Video Compression Schemes for H.264/AVC Platform*

We the below signed, after checking the dissertation mentioned above and the official record book (s) of the student, hereby state our approval of the dissertation submitted in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Electronics and Communication Engineering* at *National Institute of Technology Rourkela (India)*. We are satisfied with the volume, quality, correctness, and originality of the work.

Sukadev Meher Supervisor	Debiprasad Priyabrata Acharya Member, DSC
Samit Ari Member, DSC	Prasanna Kumar Sahu Member, DSC
External Examiner	Chairperson, DSC

Head of the Department

Department of Electronics and Communication Engineering
National Institute of Technology Rourkela (India)

Prof. Sukadev Meher
Professor

January 10, 2017

Supervisor's Certificate

This is to certify that the work presented in the dissertation titled, *Fast and Efficient Foveated Video Compression Schemes for H.264/AVC Platform*, submitted by *Deepak Singh*, Roll Number 511EC105, is a record of original research carried out by him under my supervision and guidance in partial fulfillment of the requirements of the degree of *Doctor of Philosophy* in *Electronics and Communication Engineering*. To the best of my knowledge, no significant part of the claimed research outcome embodied in it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

_____
Sukadev Meher

Dedication

Dedicated
to
my family....

Signature

Declaration of Originality

I, *Deepak Singh*, Roll Number *511EC105* hereby declare that this dissertation titled, *Fast and Efficient Foveated Video Compression Schemes for H.264/AVC Platform*, presents my original work carried out as a doctoral student of NIT Rourkela and, to the best of my knowledge, contains no material previously published or written by another person, nor any material presented by me for the award of any degree or diploma of NIT Rourkela or any other institution. Any contribution made to this research by others, with whom I have worked at NIT Rourkela or elsewhere, is explicitly acknowledged in the dissertation. Works of other authors cited in this dissertation have been duly acknowledged under the sections "Reference" or "Bibliography". I have also submitted my original research records to the scrutiny committee for evaluation of my dissertation.

I am fully aware that in case of any non-compliance detected in future, the Senate of NIT Rourkela may withdraw the degree awarded to me on the basis of the present dissertation.

January 10, 2017
NIT Rourkela

Deepak Singh

Acknowledgment

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or other contributed and extended their valuable assistance in course of this study.

I wish to express my sincere gratitude to my supervisor, Prof. Sukadev Meher, for his guidance, encouragement and support throughout this research work. His impressive knowledge, technical skills and human qualities have been a source of inspiration and a model for me to follow.

I am very much thankful to Prof. Kamalakanta Mahapatra, Head of the Department, Electronics Communication Engineering, for his constant support. I also gratefully thank my Doctoral Scrutiny Members, Prof. Debiprasad Priyabrata Acharya, Prof. Samit Ari and Prof. Prasanna Kumar Sahu for their valuable suggestions on this dissertation.

I would also like to thank fellow research colleagues for their accompaniment. It gives me a sense of happiness to be with them. Finally, I would like to thank my family and friends, whose faith and patience had always been a great source of inspiration to me.

June 29, 2016
NIT Rourkela

Deepak Singh
Roll Number: 511EC105

Abstract

Some fast and efficient foveated video compression schemes for H.264/AVC platform are presented in this dissertation. The exponential growth in networking technologies and widespread use of video content based multimedia information over internet for mass communication applications like social networking, e-commerce and education have promoted the development of video coding to a great extent. Recently, foveated imaging based image or video compression schemes are in high demand, as they not only match with the perception of human visual system (HVS), but also yield higher compression ratio. The important or salient regions are compressed with higher visual quality while the non-salient regions are compressed with higher compression ratio. From amongst the foveated video compression developments during the last few years, it is observed that saliency detection based foveated schemes are the keen areas of intense research. Keeping this in mind, we propose two multi-scale saliency detection schemes.

(1) Multi-scale phase spectrum based saliency detection **(FTPBSD)**;

(2) Sign-DCT multi-scale pseudo-phase spectrum based saliency detection **(SDCTPBSD)**.

In FTPBSD scheme, a saliency map is determined using phase spectrum of a given image/video with unity magnitude spectrum. On the other hand, the proposed SDCTPBSD method uses sign information of discrete cosine transform (DCT) also known as sign-DCT (SDCT). It resembles the response of receptive field neurons of HVS. A bottom-up spatio-temporal saliency map is obtained by linear weighted sum of spatial saliency map and temporal saliency map.

Based on these saliency detection techniques, foveated video compression (FVC) schemes (**FVC-FTPBSD** and **FVC-SDCTPBSD**) are developed to improve the compression performance further.

Moreover, the 2D-discrete cosine transform (2D-DCT) is widely used in various video coding standards for block based transformation of spatial data. However, for directional featured blocks, 2D-DCT offers sub-optimal performance and may not able to efficiently represent video data with fewer coefficients that deteriorates compression ratio. Various directional transform schemes are proposed in literature for efficiently encoding such directional featured blocks. However, it is observed that these directional transform schemes

suffer from many issues like 'mean weighting defect', use of a large number of DCTs and a number of scanning patterns. We propose a directional transform scheme based on direction-adaptive fixed length discrete cosine transform (DAFL-DCT) for intra-, and inter-frame to achieve higher coding efficiency in case of directional featured blocks. Furthermore, the proposed DAFL-DCT has the following two encoding modes.

(1) Direction-adaptive fixed length — high efficiency **(DAFL-HE)** mode for higher compression performance;

(2) Direction-adaptive fixed length — low complexity **(DAFL-LC)** mode for low complexity with a fair compression ratio.

On the other hand, motion estimation (ME) exploits temporal correlation between video frames and yields significant improvement in compression ratio while sustaining high visual quality in video coding. Block-matching motion estimation (BMME) is the most popular approach due to its simplicity and efficiency. However, the real-world video sequences may contain slow, medium and/or fast motion activities. Further, a single search pattern does not prove efficient in finding best matched block for all motion types. In addition, it is observed that most of the BMME schemes are based on uni-modal error surface. Nevertheless, real-world video sequences may exhibit a large number of local minima available within a search window and thus possess multi-modal error surface (MES). Hence, the following two uni-modal error surface based and multi-modal error surface based motion estimation schemes are developed.

(1) Direction-adaptive motion estimation **(DAME)** scheme;

(2) Pattern-based modified particle swarm optimization motion estimation **(PMPSO-ME)** scheme.

Subsequently, various fast and efficient foveated video compression schemes are developed with combination of these schemes to improve the video coding performance further while maintaining high visual quality to salient regions.

All schemes are incorporated into the H.264/AVC video coding platform. Various experiments have been carried out on H.264/AVC joint model reference software (version **JM 18.6**). Computing various benchmark metrics, the proposed schemes are compared with other existing competitive schemes in terms of rate-distortion curves, Bjontegaard metrics (BD-PSNR, BD-SSIM and BD-bitrate), encoding time, number of search points and subjective evaluation to derive an overall conclusion.

Keywords: Block matching motion estimation (BMME); Direction adaptive transform; Discrete cosine transform (DCT); Foveated video compression (FVC); Human vision system (HVS); Motion estimation (ME); Saliency detection; Video coding.

Contents

List of Figures

List of Tables

List of Acronyms

1D	One-dimensional
2D	Two-dimensional
2D-DCT	Two-dimensional Discrete cosine transform
1D-DCT	One-dimensional Discrete cosine transform
4CIF	4 times Common Intermediate Format resolution (704×576 pixels)
AUC	Area under the curve
BD-bitrate	Bjontegaard delta bit-rate
BD-PSNR	Bjontegaard delta PSNR
BD-SSIM	Bjontegaard delta SSIM
BMME	Block matching motion estimation
CABAC	Context adaptive binary arithmetic coding
CAVLC	Context adaptive variable length coding
CIF	Common Intermediate Format of resolution: 352×288 pixels
CPSNR	Composite peak signal to noise ratio
CPSO	convetional PSO
CSP	Cross-search pattern
DAFL-DCT	Direction-adaptive fixed length-DCT
DAFL-HE	Direction-adaptive fixed length —high efficiency
DAFL-LC	Direction-adaptive fixed length —low complexity
DAME	Direction-adaptive motion estimation
DART	Direction-adaptive residual transform
DCT	Discrete cosine transform
DDCT	Directional discrete cosine transform
DHS	Diamond and hexagon search
DS	Diamond search
DST	Discrete sine transform
EPZS	Enhanced predictive zonal search
ES	Exhaustive search
FFS	Fast full search
FPS	Frames per second
FP-BMME	Fixed pattern based BMME

FS	Full search
FTPBSD	Multi-scale phase spectrum based saliency detection
FVC	Foveated video compression
HD 720p	High definition progressive format of 1280×720 resolution
HS	Harmony search
HEVC	High efficiency video coding
HEXBS	Hexagon-based search
HHSP	Horizontal hexagonal search pattern
IP	Intra-prediction
KLT	Karhunen-Loeve transform
KSP	Kite search patter
LC-BMME	Lower complexity based BMME
MC	Motion-compensation
MCP	Motion-compensated-prediction
ME	Motion estimation
MES	Multi-modal error surface
MSE	Mean of squared error
MSSIM	Mean SSIM
MV	Motion vector
MVD	Motion vector difference
MVP	Motion vector prediction
PMPSO-ME	Pattern-based modified particle swarm optimization motion estimation
PSNR	Average peak signal to noise ratio
PSO	Particle swarm optimization
QCIF	Quarter Common Intermediate Format of 176×144 resolution
QP	Quantization parameter
ROC	Receiver operating characteristics
RSP-BMME	Reduced search points based BMME
SAD	Sum of absolute difference
SATD	Sum of absolute transformed difference
SDCTPBSD	Sign-DCT multi-scale pseudo-phase spectrum based saliency detection
SDSP	Small diamond search pattern
SSIM	Structural similarity index measure
SUMH	Simple UMH
UES	Uni-modal error surface
UMHexagonS/UMH	Hybrid unsymmetrical-cross multi-hexagon-grid search
VHSP	Vertical hexagonal search pattern
ZMV	Zero motion vector

List of Symbols

$f(i,j)$	Original frame with spatial co-ordinates (i,j)
$g(i,j)$	Encoded frame
t, δ	Time (temporal index)
T	Encoding time
$\tilde{f}$	Reconstructed frame
F	Frequency domain frame
W, H	Number of rows (columns) of a video frame
r, c	Number of rows (columns) of blocks a video frame
M, N	Number of rows (columns) of a block
i, j, k, l	Spatial co-ordinates
A'	Transpose of a matrix A or a vector
A^*	Complex conjugate of A
SM	Saliency Map
N_L	Number of levels for Gaussian image pyramid
σ	Standard deviation
$\mathscr{F}$	Fourier Transform
$\mathscr{F}^{-1}$	Inverse Fourier Transform
ω	Weighting factor
$D_E(i,j)$	Euclidean distance at co-ordinate (i,j)
ρ	Correlation coefficient
$C(u)$	Weighting factor at u
E_v	Expectation value
QP	Quantization parameter
J	Rate-distortion (RD) cost
D_s	Distortion
R	Bit-Rate
λ	Lagrangian multiplier
ϖ	Directional DAFL-DCT transform modes
Υ_{SAD}	Threshold (SAD)
Υ_{MSE}	Threshold (MSE)
W_s	Size of search window

$\overrightarrow{\mathrm{mv}}$	Motion vector
$\overrightarrow{\mathrm{i}}$	Vector
$\overrightarrow{\mathrm{zmv}}$	Zero motion vector (ZMV)
Ψ	Maximum displacement
I_w	Inertia weight
$rand$	Random number
itr	Iteration number
κ	momentum factor
Φ	Fitness function
$\mathbb{R}$	Set of real numbers
Δ	Relative change in a parameter

Chapter 1

Introduction

Preview

Recently, the exponential growth in networking technologies and widespread use of video content based multimedia information over internet for mass communication through social networking, e-commerce, education, etc. have promoted the development of video coding to a great extent. Various video coding schemes have already been designed for seamless transmission of digital video data and for mass storage of digital information. The primary goal of a video coding standard is to achieve higher compression performance while maintaining high visual quality. A human eye is space-variant non-uniform resolution sampling system. Hence, foveation based video coding yields higher compression performance by varying the visual quality of video data across the space to match the non-uniform spatial sampling of a human eye. In the present doctoral research work, efforts are made to develop fast and efficient foveated video compression schemes that achieve higher compression performance as well as higher visual quality at a lower computational complexity.

The following topics are covered in this chapter.

- Digital video
- Introduction of video compression schemes
- Performance metrics
- Problem statement
- Chapter-wise organization of thesis
- Conclusion

1.1 Digital Video

Digital video is a three-dimensional data of a dynamic visual scene, sampled spatially and temporally. A visual scene temporally sampled at any time instant is known as a frame.

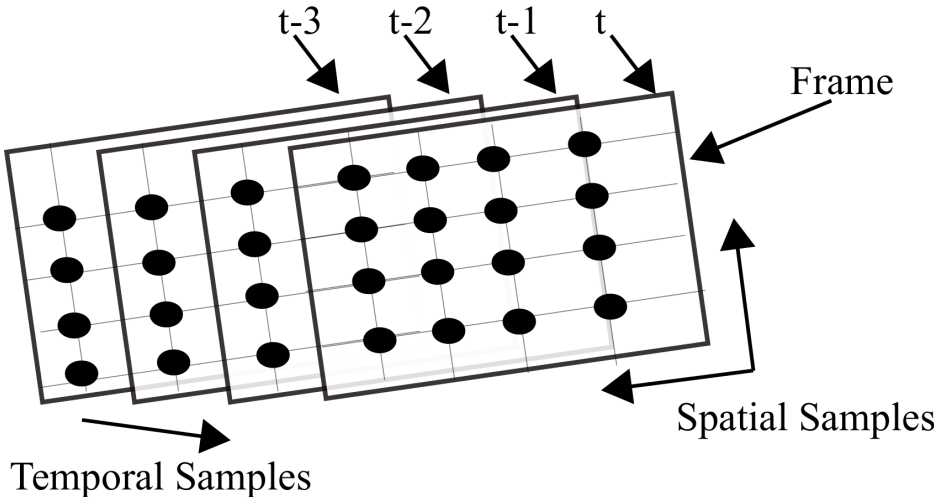


Figure 1.1: Illustration of spatio-temporal sampling of a video scene

The sampling is done repetitively and its sampling rate should not be below $1/25$ second for producing smooth moving vision effect without any jerking artefacts [1]. Figure 1.1 illustrates the spatio-temporal sampling of a scene for producing digital video. Each spatio-temporal sample is represented by a pixel $f(i, j, t)$. Every frame has a width of W pixels and height of H pixels that gives frame size as $H \times W$ pixels [2]. Each pixel has a fixed number of bits which is known as intensity-range or colour depth. More the number of bits representing a pixel, better will be the colour depth and hence better contrast.

Usually, a monochrome video is represented by 1-byte pixels, whereas a colour video by 3-byte pixels each having three colour components separately represented by one byte each. There are various colour space models that describe a colour video. The most common models used to represent digital colour video data are RGB and YCbCr [3]. In RGB colour space, each pixel comprises three numbers representing red, green and blue components. The combination of these colour components will produce any desired colour. In Figure 1.2, colour components are shown for a video frame. Figure 1.2(a) is the original image. Figure 1.2(b) represents red component, where the red colour pixels are brighter, whereas in Figure 1.2(c) which is green colour component of original frame, green colour pixels are brighter. Similarly, for Figure 1.2(d) blue colour pixels are brighter than others. The RGB colour space model is mostly used in computer graphics and rarely used for real-world examples. Since all the three primary colour components are equally important to represent a colour, the storage requirement for an RGB colour frame is three times that of a monochrome frame.

YCbCr (or equivalently, YUV) is an efficient alternative colour space model for a video data. It is known that the human visual system (HVS) is highly sensitive to luminance (intensity) than colour [4]. Hence, a colour video can be stored efficiently by extracting luminance and representing it with higher resolution than the colour components. In YCbCr, Y represents luminance while Cb and Cr represent red and blue colour differences,

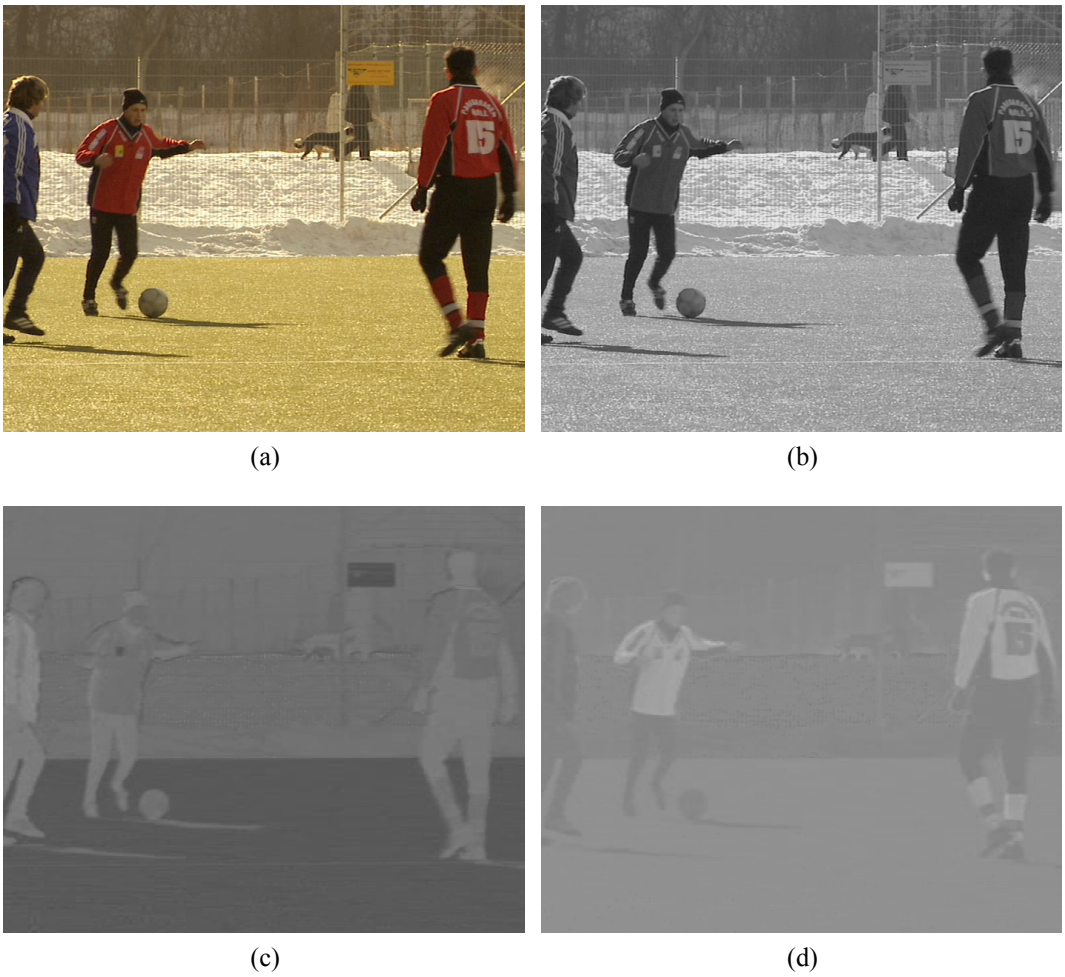
(a) (b)

(c) (d)

Figure 1.2: RGB colour components of *Soccer* video frame: (a) original, (b) red component, (c) green component and (d) blue component

respectively. Figure 1.3 shows YCbCr components for a video frame. In Figure 1.3(c) and Figure 1.3(d), the colour differences Cb and Cr are shown with dark to light from negative differences to positive differences. For general purpose, 4:2:0 sampling format is used for YCbCr video data [3, 5, 6]. In 4:2:0 sampling format, Y will be of full resolution while Cb and Cr will have half horizontal and half vertical resolutions compared to Y component. In other words, there will be only one component of Cb and Cr for 2×2 Y components. This reduces the storage and processing requirement of video data considerably as compared to RGB colour space without losing significant visual quality. The colour space conversion [7] from RGB to YCbCr and vice versa are given by:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B \\ Cb &= 0.564\,(B - Y) \\ Cr &= 0.713\,(R - Y) \end{aligned} \tag{1.1}$$



(a) (b)

(c) (d)

Figure 1.3: YCbCr colour components of *Soccer* video frame: (a) original, (b) Y-component, (c) Cb-component and (d) Cr-component

$$
\begin{aligned}
R &= Y + 1.402Cr \\
G &= Y - 0.344Cb - 0.714Cr \\
B &= Y + 1.772Cb
\end{aligned}
\tag{1.2}
$$

In the present research work, YCbCr video sequences are taken as input source. The fundamentals of video compression schemes are discussed in the following section.

1.2 Fundamentals of Video Compression

1.2.1 Background

In the modern world, the demand of video data has increased manifold due to massive internet application like social networking, e-governance, security and surveillance, video telephony. Hence, the network bandwidth has become a major bottleneck for efficient

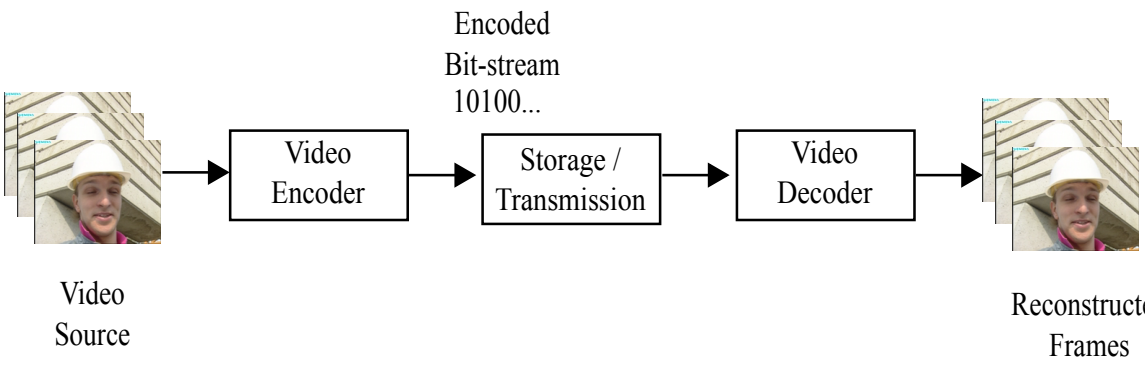


Figure 1.4: Typical video coding system

transmission of these vast amount of video data in real-time even if the present technology offers quite large bandwidths. Most probably, this problem will continue for ever since the modern human civilization will demand more and more for video transmission applications in future. Therefore, a well designed and efficient video compression system is always required to reduce transmission bit-rate for a video data content without degrading the visual quality significantly. In a heterogeneous network, where medium to low data rates are supported, transmission of video data is even a more challenging task. The data rates available within a network vary across the channels according to the characteristics of a network, i.e. the types of the transmission channel and the receiving data terminal as well as the network traffic congestion. Consequently, video data must be transmitted at a variety of bit-rates to have efficient transmission. Some efficient and adaptive video compression schemes are required to solve these issues [8–10]. A typical video coding system is shown in Figure 1.4. A video data generated at the source is encoded with low bit-rate by a video encoder. The compressed video data is either sent to storage devices or transmitted through a communication channel. At the receiving end, the compressed video data is decoded by a video decoder and reconstructed video frames are displayed to users.

There are two types of compression schemes: lossless and lossy. In a lossless compression scheme, the video data is represented by less number of bits without any loss of information. Hence, lossless compression scheme achieves a perfect reconstruction of an original information after decompression. However, a lossy compression scheme yields higher compression performance as compared to its counterpart but at the cost of some loss of information up to an acceptable limit [2]. The compression ratio of an encoder is defined as:

$$\text{Compresson ratio} = \frac{\text{Number of bits in uncompressed data}}{\text{Number of bits in compressed data}} \tag{1.3}$$

Various video coding schemes have been developed and standardized by two international study groups in the last two decades. One is video coding expert group (VCEG) of international telecommunication union-telecommunication (ITU-T) [11] and another is moving picture expert group (MPEG) of international organization for standardization

(ISO) and the international electrotechnical commission (IEC) [12]. In 1990, the ITU-T has adopted H.261 video standard with the aim of transmitting a video data over integrated services digital network (ISDN) for video conferencing and video telephony applications [13]. Later, in 1993, ISO/IEC has adopted MPEG-1 video standard for storage devices with a target bit-rate of 1.5Mbps for compact disc [14]. In 1995, VCEG and MPEG groups jointly finalized a video standard, known as MPEG-2 by ISO/IEC [15] and recommendation H.262 by ITU-T H.262 1995 [16]. The MPEG-2 has been developed for storage on digital versatile disc (DVD) or for video on demand (VOD) standard definition (SD) and high-definition (HD) digital television broadcasting with target bit-rates of $4-15$ Mbps. H.263 has been finalized by ITU-T in 1996 for video telephony application over circuit and packet switched network from low bit-rates to higher bit-rates [17]. For a wide range of applications like object-based coding [18], encoding of natural and/or synthesized video objects [19], MPEG-4 part 2 is adopted in 1998 by ISO [20].

In 2005, the joint video team (JVT), a combined team of VCEG and MPEG, has introduced advanced video coding (AVC) which is also known as H.264 by ITU-T and MPEG-4 part 10 by ISO [6]. H.264/AVC yields higher compression performance, approximately 50% more than MPEG-2 with the cost of higher computational complexity. Recently, high efficiency video coding (HEVC) has been adopted by ITU-T and ISO, which is developed by the joint collaborative team on video coding (JCT-VC) of VCEG and MPEG experts [21]. HEVC yields highest compression performance, but at the expense of very high computational complexity as compared to H.264/AVC. Presently, H.264/AVC is being widely used in streaming internet resources, web application software, video telephony, high-definition television (HDTV) broadcasting, digital cinema format and many more. However, the HEVC performs better in high-resolution videos than in low-resolution videos meeting its design goals. It is observed that HEVC is best for low bit-rate applications but it is not suitable for low delay broadcasting applications due to its higher complexity [22–24]. Many services, with real-time applications in today's video communication run on battery operated mobile devices and employ the H.264/AVC in their video related applications than HEVC as they can not tolerate a significant amount of delay and complexity in coding due to limited resources. Hence, we have given more emphasis on H.264/AVC than HEVC. In the present research work, we have chosen H.264/AVC as video coding platform for analysing the performance of our proposed schemes. The detailed discussion on H.264/AVC architecture is given in the next section.

1.2.2 Architecture of H.264/AVC

H.264, also known as advanced video coding, (ISO designates it MPEG-4 part 10) is an efficient video compression scheme. It provides higher compression performance and robust transmission than its predecessors. There are many profiles of H.264, which define a set of tools that target a specific class of applications ranging from video conferencing and mobile

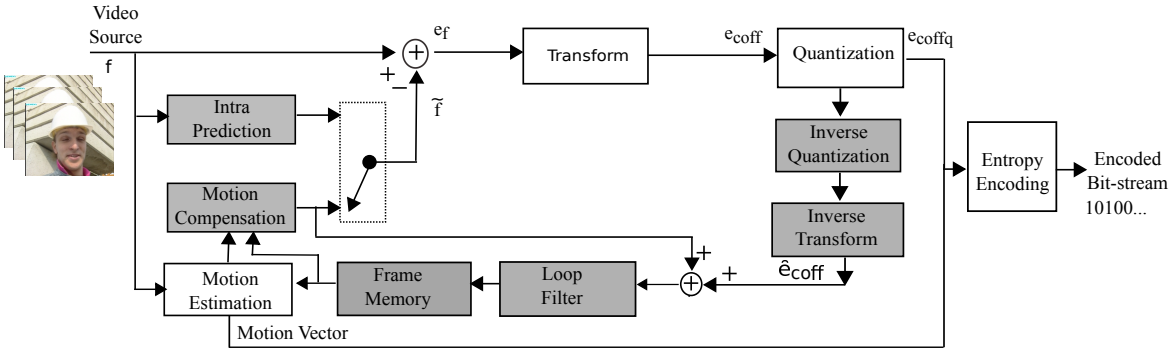


Figure 1.5: Block diagram of H.264/AVC video encoder

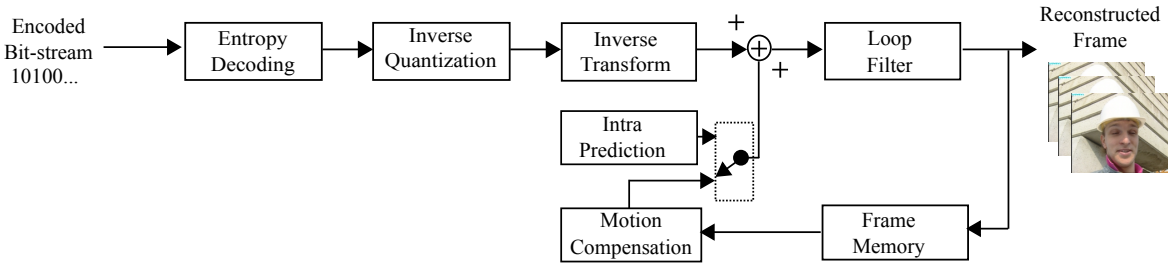


Figure 1.6: Block diagram of H.264/AVC video decoder

video applications to blu-ray disc storage and HDTV broadcasting over fixed and wireless networks with different transport protocols [25]. The encoder and decoder of H.264/AVC are shown in Figure 1.5 and Figure 1.6 respectively. In Figure 1.5, gray blocks represent in-built H.264/AVC video decoder. The various functional elements which make H.264/AVC an efficient video compression schemes are discussed below.

Slices and macroblocks

In H.264/AVC, a video sequence consists of many video pictures. A picture can be a frame or a field. A video picture is divided into macroblocks. Each macroblock consists of one 16×16 samples of Y component and two blocks of Cb and Cr components. H.264/AVC supports slice architecture, where each video picture is encoded as one or more slices [26]. Each slice contains an integral number of macroblocks. It may vary from a single macroblock to the whole picture. The slice can be encoded and decoded independently. There are five types of slices supported in H.264/AVC, which are I-, P-, B-, SI-, and SP-slices [8]. In an I-slice, all macroblocks are encoded without any reference to other frames whereas in P-slice and B-slice macroblocks other than intra macroblocks are encoded with the help reference frames. The SI- and SP-slices are switching slices and used for switching between two bit-streams [5].

Intra-prediction

In intra-coding, the macorblocks are predicted from the current frame only and errors are encoded. This improves intra-coding compression performance significantly. H.264/AVC

supports nine intra-prediction modes for 8×8 and 4×4 each and four intra-prediction modes for 16×16 luma component and 8×8 chorma components [6].

Inter-prediction

The H.264 supports 7 types of blocks with dimension of 16×16, 16×8, 8×16, 8×8 , 8×4, 4×8 and 4×4 pixels for inter-prediction [5]. Smaller the block size, better is the prediction. Hence, smaller blocks are preferred for a high detail area. Each block is predicted from a reference picture and displacement is given by motion vectors. The precision of a motion vector is of quarter-pixel for luma components and $1/8$-pixel for chroma components [6]. The H.264/AVC also supports multiple reference motion compensation with up to 16 reference frames in contrast to previous video coding standards which supported only one reference frame [25].

Transform and quantization

The H.264/AVC supports multiple block size multiplier-free integer transforms for prediction residuals. The 4×4 Hadamard transform is applied to luma DC-coefficient block if intra 16×16 mode is selected. Similarly, a 2×2 Hadamard transform is used for chroma DC-coefficients blocks and for other residual blocks 4×4 integer transform, 8×8 integer transform or both are applied depending on the selected transform mode [25]. The H.264/AVC uses scaler quantizer for all transform coefficients. The quantizer value is selected using quantization parameter (QP) which can have 52 values [26].

Entropy coding

The H.264/AVC supports two types of entropy coding schemes: context adaptive variable length coding (CAVLC) and context adaptive binary arithmetic coding (CABAC) [5]. For low complexity, CAVLC is selected whereas for higher compression, a more complex encoding scheme, CABAC is employed. CABAC assigns a non-integer number of bits for each variable rather than integer number of bits by variable-length coding [27].

In-loop deblocking filter

The basic limitation of a block transform based video coding scheme is blocking artefacts. Since the block edges are less accurately reconstructed than the inner pixels of a block, the blocking artefacts are visible at boundary edges [1]. The H.264/AVC applies an adaptive deblocking filter to mitigate these blocking artefacts. In previous video coding standards, the deblocking filter is used as post-processing filter; but in H.264/AVC, the deblocking filtering is carried out in encoding loop for achieving higher visual quality when video frames are reconstructed at the decoder end [25].

1.3 Performance Metrics

The performance of a video coding scheme is evaluated based on subjective and objective qualities. In the subjective quality evaluation, the visual quality of a reconstructed video frame is observed by a human expert [28]. The difficulty of this method is the perceptibility about visual quality not only varies from person to person, but also with targeted applications. However, the objective quality evaluations are based on distortion or error related parameters derived mathematically [29]. The objective quality evaluations are more accurate and repeatable. Performance metrics are defined for video compression schemes and saliency detection schemes in the subsequent sections.

1.3.1 Performance metrics for video compression schemes

The primary goal of a video compression scheme is to represent a video data in a compact form while preserving the visual quality as far as possible. Compression ratio is one of the principal parameter of a compression scheme and it is calculated by (1.3), but it does not give sufficient information regarding the compression scheme. An efficient low bit-rate compression scheme not only achieves higher compression ratio, but also yields lower distortion in visual quality. Therefore, various distortion based performance metrics are present in literature to evaluate the performance of a video coding such as sum of absolute difference (SAD), sum of squared difference (SSD), peak signal to noise ratio (PSNR), structural similarity index (SSIM). Among these distortion metrics, some are used to find the performance of the proposed video compression schemes. They are described in detail in the following section.

Let an original video frame and the reconstructed video frame are represented by $f(i,j)$ and $\tilde{f}(i,j)$ respectively. Here, i and j represent the spatial co-ordinates of the digital video frame. The video frame size be $H \times W$ pixels, i.e $i = 1, 2, \cdots, H$ and $j = 1, 2, \cdots, W$. The SAD and SSD are defined as:

$$SAD = \sum_{i=1}^{H} \sum_{j=1}^{W} \left| \tilde{f}(i,j) - f(i,j) \right| \tag{1.4}$$

$$SSD = \sum_{i=1}^{H} \sum_{j=1}^{W} \left(\tilde{f}(i,j) - f(i,j) \right)^2 \tag{1.5}$$

Higher value of SAD represents lower visual quality. It is the same for SSD. But, SAD is simple and fast computed distortion metric than SSD [1]. PSNR is the ratio of peak signal power to peak noise power and it is defined using logarithmic scale in dB. If a pixel of a video frame is represent by 8-bit value, then the maximum value of a pixel is 255 [30]. Hence the PSNR is defined as:

$$PSNR = 10log_{10}\Big(\frac{255^2}{MSE}\Big) \tag{1.6}$$

where MSE (mean of absolute error) is calculated as:

$$MSE = \frac{1}{W \times H}\sum_{i=1}^{H}\sum_{j=1}^{W}\big(\tilde{f}(i,j) - f(i,j)\big)^2 \tag{1.7}$$

For normal to high quality video, the PSNR varies around 30 dB to 50 dB [31]. For a colour video frame that has three colour components Y, Cb and Cr, another metric, composite peak signal to noise ratio (CPSNR) in dB is used [32]. It is defined as:

$$CPSNR = 10log_{10}\Big(\frac{255^2}{\frac{1}{3}(MSE_Y + MSE_{Cb} + MSE_{Cr})}\Big) \tag{1.8}$$

where MSE_Y, MSE_{Cb} and MSE_{Cr} represent the MSE values of Y, Cb and Cr components respectively.

Though these performance metrics based on PSNR are widely popular for evaluating the efficiency of video compression schemes, they do not give true indication of the distortion introduced by compression schemes to achieve higher compression efficiency. In addition to these performance metrics, structural similarity index measure (SSIM) is used as distortion measure to evaluate the distortions in reconstructed video frames due to compression. The SSIM is based on HVS characteristics. It is known that the HVS is more adaptive to extract structural information from a visual scene than error between two pixels. Therefore distortion in structural information is a good measure of finding the similarity between two video frames [33]. The SSIM is a window based approach i.e. SSIM is calculated for each block typically of 8×8 pixels size. Though SSIM lies in the range of $[-1, 1]$, it is mostly given in the interval of $[0, 1]$. The closer value towards 0 indicates lower visual quality while higher picture quality yields SSIM value nearer to 1. The SSIM is a combination of three factors: local luminance difference, local contrast difference and local structure difference. Moreover, these factors are relatively independent and do not affect each other [34]. The SSIM is calculated as:

$$SSIM = \sum_{i=1}^{M}\sum_{j=1}^{N}\Big(\frac{2\mu_f\mu_{\tilde{f}} + C_1}{\mu_f^2 + \mu_{\tilde{f}}^2 + C_1}\,\frac{2\sigma_f\sigma_{\tilde{f}} + C_2}{\sigma_f^2 + \sigma_{\tilde{f}}^2 + C_2}\,\frac{\sigma_{f\tilde{f}} + C_3}{\sigma_f\sigma_{\tilde{f}} + C_3}\Big) \tag{1.9}$$

where M and N are the number of rows and columns of pixels in a block, μ_f and $\mu_{\tilde{f}}$ are the respective local pixels mean of $f(i,j)$ and $\tilde{f}(i,j)$, σ_f and $\sigma_{\tilde{f}}$ are the respective local pixel standard deviations of $f(i,j)$ and $\tilde{f}(i,j)$ and $\sigma_{f\tilde{f}}$ is the covariance of $f(i,j)$ and $\tilde{f}(i,j)$ after removing their means. The coefficients C_1, C_2 and C_3 are small positive constants employed to numerical instability [35].

Since the SSIM index is defined for a block, the overall SSIM value for a single video frame is measured by mean SSIM (MSSIM) value that is defined as:

$$MSSIM = \frac{1}{r \times c} \sum_{i=1}^{r} \sum_{j=1}^{c} SSIM(i,j) \tag{1.10}$$

where r and c are number of rows and columns of blocks in a single video frame.

Recently, the Bjontegaard metrics, Bjontegaard delta bit-rate (BD-bitrate) and Bjontegaard delta PSNR (BD-PSNR) are gaining much popularity as benchmark metrics to evaluate coding efficiency of a scheme with respect to another. Bjontegaard metrics calculate the average bit-rate or PSNR difference between two encoders' rate-distortion (R-D) curves which represent the relations between PSNR obtained by encoding the video data with different bit-rates [36]. The BD-PSNR represents the average PSNR difference in dB for the same bit-rate and BD-bitrate corresponds to average bit-rate difference in percentage for the same PSNR. In Bjontegaard metric, positive numbers in BD-PSNR represent PSNR gain, while negative numbers in BD-bitrate show reduction in bit-rate and vice-versa. In addition, we have also included BD-SSIM which represents the average SSIM difference of two video encoders for the same bit-rate.

1.3.2 Performance metrics for saliency detection techniques

Let an original video frame and the saliency map of that video frame are represented by $f(i,j)$ and $SM(i,j)$ respectively. Here, i and j represent the spatial co-ordinates of the video frame. The video frame size be $H \times W$ pixels. An object map (o_b) is generated by appropriate thresholding of the saliency map $SM(i,j)$ of size $H \times W$ pixels for a binary map outcome. g_b is the ground truth of saliency map, which is already in binary form.

Precision represents a fraction amount of correctly detected salient objects, while recall measures a fraction of ground truth detected as salient objects. F-measure corresponds to a weighted harmonic mean of precision and recall with a non-negative value of α. The precision, recall and F-measure are mathematically calculated as:

$$Precision = \frac{\sum_{i=1}^{W} \sum_{j=1}^{H} g_b(i,j) o_b(i,j)}{\sum_{i=1}^{W} \sum_{j=1}^{H} o_b(i,j)} \tag{1.11}$$

$$Recall = \frac{\sum_{i=1}^{W} \sum_{j=1}^{H} g_b(i,j) o_b(i,j)}{\sum_{i=1}^{W} \sum_{j=1}^{H} g_b(i,j)} \tag{1.12}$$

$$F - measure = \frac{(1+\alpha) \times Precision \times Recall}{\alpha \times Precision + Recall}, \quad \alpha = 0.5 \tag{1.13}$$

In special case of precision$= 0$ and recall$= 0$ then F-measure$= 0$.

The precision, recall and F-measure reach maximum value of 1 if and only if o_b equals to g_b.

Receiver operating characteristics (ROC) is another benchmark metric for performance evaluation of a decision system. It represents the trade-off between true hit rate and false alarm rate of a decision system. In case of saliency detection, the ROC curve measures accuracy of predictions of fixation and non-fixation regions based on bottom-up saliency detection methods [37]. The ROC curve is defined as a plot between true positive rate (TPR) or true hit rate in the y-axis versus false positive rate (FPR) or false alarm rate in the x-axis for different threshold values. So, each point on curve represents values of TPR and FPR at various decision thresholds. TPR (also known as recall or sensitivity) is defined as a fraction of true fixation points that comes into fixation points obtained by a saliency map as a result. However, FPR (also known as $(1 - specificity)$) is defined as a fraction of true non-fixation points comes into fixation points obtained by a saliency map. The values of TPR and FPR are calculated by following equations:

$$TPR = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} g_b(i,j)o_b(i,j)}{\sum_{i=1}^{H}\sum_{j=1}^{W} g_b(i,j)} \tag{1.14}$$

$$FPR = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W} \tilde{g}_b(i,j)o_b(i,j)}{\sum_{i=1}^{H}\sum_{j=1}^{W} \tilde{g}_b(i,j)} \tag{1.15}$$

where g_b depicts ground truth, o_b shows object map and $\tilde{g_b}$ represents complement of g_b depicting background points.

A measure of overall performance of ROC curve is area under the curve (AUC). The AUC is a combined measure of sensitivity and specificity. As both the axis have value from 0 to 1, the value of AUC also lies between 0 to 1. A perfect accurate system has AUC equal to 1. In other words, a system performance will be considered as superior, if it has AUC value closer to 1 [38]. In saliency detection, the AUC measures the prediction of fixation points of a human eye by saliency maps. The chance performance system has ambiguity in decision accuracy for fixation points and has AUC equal to 0.5 which is a practical lower limit for performance.

1.4 Problem Statement

A video data contents huge amount of information and storing or transmitting these enormous data is a very challenging task, specifically at heterogeneous network. Hence, the video coding is a prominent area of research due to its vast applications in wired or wireless network and low cost handheld devices with less storage and computing capacity. Researchers have developed many compression schemes for low-bit rate applications [39–41]. But these schemes yield poor visual quality for high compression and vice-versa. In addition, foveated video coding scheme that achieves non-uniform resolution of video coding by prioritizing the visual scene according to the characteristics of HVS, improves the compression efficiency considerably. Based on thorough investigation, it is observed that there exists a scope for further improvement in video compression scheme to yield higher compression efficiency and higher visual quality as well. The video compression schemes to be developed must have low computational complexity, so that they will be easily accommodated to existing video coding standards for real-time applications. Recently, foveated video compression schemes are widely used in low to medium bit-rate applications [42–44].

Hence, the following research problem has been taken.

Problem Statement:

To develop efficient foveated video compression schemes, for H.264/AVC platform, that yield higher compression ratio and better visual quality but with lower computational complexities for low and medium resolution applications like mobile based video telephony and conferencing, standard-definition TV broadcasting and web based video related services.

1.5 Chapter-wise Organization of Thesis

The chapter-wise organization of thesis is presented here.

Chapter 1 Introduction

Chapter 2 Literature Review

- 2.1 Foveated video compression
- 2.2 Directional transform
- 2.3 Motion estimation
- 2.4 Conclusion

Chapter 3 Development of Foveated Video Compression Schemes

- 3.1 Introduction

1.6 Conclusion

This chapter provides a brief introduction on video compression scheme. The fundamental of digital video is discussed. The background of video compression schemes and architecture of H.264/AVC video coding standard are briefly analysed. The performance metrics for evaluating the efficiency of saliency detection techniques and video compression schemes are also described. Observing the shortcomings of existing schemes in the literature, a research problem is formulated and stated explicitly. Finally, chapter-wise organization of the dissertation is presented.

Chapter 2

Literature Review

Preview

A space-variant non-uniform resolution image can be generated by various foveation filtering schemes. The encoding of oblique featured video data is a challenging task. Different directional transform schemes are available in literature, which efficiently encode these oblique featured video data. Motion estimation is one of the very important tools of a hybrid video compression schemes. Various motion estimation schemes are present in literature to find out the best matched block in a reference frame and enhance the compression efficiency with minimum computation cost. In this chapter, some well-known, efficient, standard and benchmark schemes related to different tools of efficient foveated video compression schemes, are studied. The proposed schemes, developed and designed in this doctoral research work, are compared against these in subsequent chapters. Therefore, attempts are made here for a detailed and critical analysis of these schemes.

The following topics are covered in this chapter.

- Foveated video compression
- Directional transform
- Motion estimation
- Conclusion

The literature review is categorized into three domains of the proposed foveated video compression schemes as shown in Figure 2.1. The detailed discussion of each category is given below.

2.1 Foveated Video Compression

Recently, foveated video compression (FVC) schemes have gain major interest by many researchers in the field of video coding. Since FVC schemes exploit non-uniformity in the resolution of the retina by allocating more number of bits to visual fixation points and reducing resolution drastically away from the fixation points, it delivers perceptually high

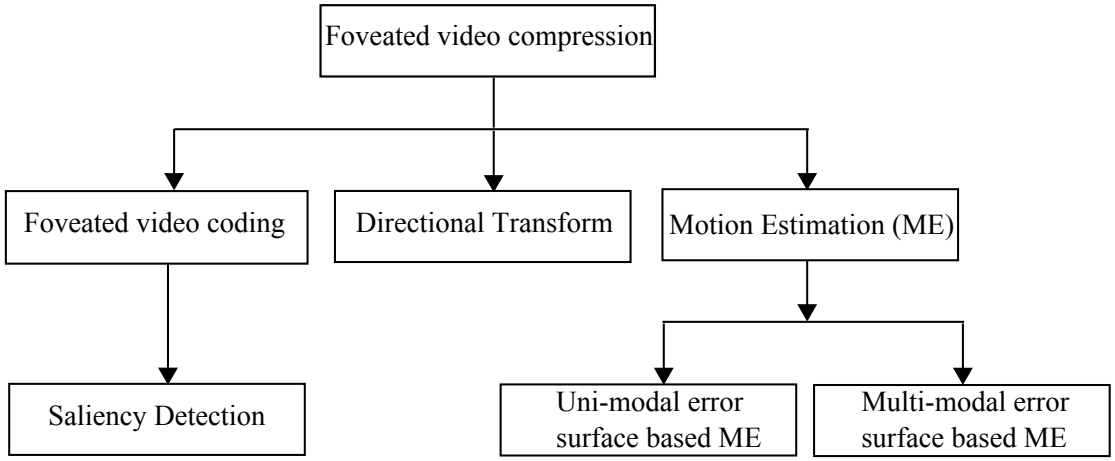


Figure 2.1: Categorisation of literature review

quality at greatly reduced bandwidths. There are several efficient foveated video processing schemes available in literature, for example, foveation filtering (local bandwidth reduction) [45], saliency detection based foveating [46–48] and wavelet based foveated compression [43]. In 1993, Silsbee et al. have introduced the image coding based on the properties of human visual system (HVS) [49]. The video is encoded by dividing the frame into a number of spatio-temporal patterns which are based on spatio-temporal properties of HVS. The adaptation of foveated processing to various video coding standards is demonstrated by [45, 50–52].

Broadly, foveation method can be classified into three categories:

1. geometry based foveation (GBF),
2. filtering based foveation (FBF) and
3. multi-resolution based foveation (MBF).

In GBF schemes, uniformly sampled image coordinates are transformed into spatial variant coordinates by logmap transform, also known as foveation coordinate transform, which exploits the retina sampling geometry [53–57]. Wallace et al. [53] and Kortum and Geisler [54] have shown geometric transformation of uniform sampled image to non-uniform space variant sampled image using superpixel. The superpixels are generated to match the retinal sampling distribution by grouping and averaging the uniform pixels. Lee and Bovik have shown that foveation is a coordinate transformation from cartesian coordinates to curvilinear coordinates and a local bandwidth is uniformly distributed over curvilinear coordinates for a foveated image [55]. Similarly, Azizi et al. have proposed region selective image compression based on warping the desired non-uniform sampling to uniform lattice using circular spatial warping algorithm [56]. Major issues with GBF are shifting from integer position to non-integer and blocking effects in superpixel boundaries. Hence, additional computations are required to overcome these constraints but at the cost of higher computational complexity.

Table 2.1: Summary of literature survey related to foveated video compression

Year	Authors	Approach	Ref.
1993	Silsbee et al.	Spatio-temporal patterns properties of HVS	[49]
1994	Wallace et al.	Logmap transformation	[53]
1996	Kortum and Geisler	Geometric transformation using super-pixel	[54]
1998	Geisler and Perry	Multi-resolution based foveatation	[58]
1999	Lee et al.	DCT based low-pass filtering	[59]
2000	Lee and Bovik	Geometric transformation using curvilinear coordinates	[55]
2000	Azizi et al.	Geometric transformation	[56]
2001	Lee et al.	DCT based low-pass filtering	[52]
2003	Wang et al.	Wavelet based low-pass filtering	[60]
2003	Lee and Bovik	Wavelet based low-pass filtering	[45]
2007	Rosenbaum and Schumann	Wavelet based multi-resolution	[42]
2010	Galan et al.	Wavelet based low-pass filtering	[43]
2013	Shang et al.	DCT based low-pass filtering	[61]
2015	Chessa and Solari	Log-polar transformation	[57]

In FBF methods, foveation is achieved by removing high frequencies of the image with low-pass filter where the cut-off frequency depends on local retinal sampling density. Since retinal sampling is space-variant in nature, a filter bank of low-pass filter is required to generate a foveated image [43, 45, 52, 59, 60]. The FBF is implemented either by discrete cosine transform (DCT) [52, 59], or wavelet transform [43, 45, 60]. In DCT domain, FBF may be combined to transform and quantization module of a video coding scheme. But, wavelet transform generates frequency subbands and therefore FBF is employed as a preprocessing task to standard video codecs.

The MBF schemes are technically a hybrid approach of geometric transformation and filtering process. In MBF, a pyramid structure is generated by down-sampling the uniform sampled image with different scales. Gaussian pyramid or Laplacian pyramid method can be used to generate a multi-resolution structure [62]. Subsequently, filtering process is employed at each scale of the uniformly sampled image to achieve foveation. Geisler and Perry have demonstrated applications of MBF system for real-time image and video coding [58]. The wavelet transform is also used to create multi-resolution images and feature maps are extracted from these multi-scale maps to generate foveated image in image compression scheme such as JPEG2000 [42].

The summarized literature survey related to FVC is given in Table 2.1.

2.1.1 Saliency detection

In human visual system (HVS), resolution varies spatially across the scene with higher resolution towards fovea, which is centre of the eye's retina and decreases rapidly towards the periphery of eyes. Therefore, selective visual attention is governed by saccadic movement of eyes towards fovea [63]. Visual selection decides the fixation and non-fixation points in a scene. Another approach is based on processing visual information selectively. In this approach, the visual selection is based on some attributes or features of a scene, which include intensity, colour, orientation, motion direction, velocity, shape and some other properties. A region or object which is different from its surrounding gets higher attention and is known as salient region or object. In FVC schemes, salient regions are the foveated points which will have higher visual quality than non-salient regions.

There have been many saliency detection methods available in the literature. However, the first biological plausible architecture is proposed by Itti et al.[64]. In Itti's model (IM), a hierarchical nine sub-scaled Gaussian pyramid model combines various low level features computed by 'centre-surround' operations similar to receptive field operations of HVS. A set of feature maps, based upon intensity, colour and orientation, are determined and all fed to a master saliency map. However, highly parametric approach and intensive computational cost are the major bottlenecks of this model. A system called Neuromorphic Vision C++ Toolkit (NVT) is developed based on Itti's proposed model. Later, based on Rensink's theory of change blindness [65], Walther has created the most useful commercial product SaliencyToolBox (STB), for determining the fixation points for visual attention [66].

Ma et al. have used contrast based (CB) feature map to determine saliency [67]. They have shown that contrast is the most important feature which directs the human visual attention more than by any other feature like colour, texture or orientation. Liu et al. have constructed a scale-invariant saliency map for an image through a multi-scale block-level pixel based contrast localization [68]. They have proposed to divide the image into regions for enhancing the saliency map with the region based information. But, it is found that the proposed method may mislead the outcome for high contrast edges.

Bruce et al. have proposed a model of bottom-up saliency based on the principle of maximizing information sampled from a scene, computed as Shannon's self-information [37]. Harel et al. have determined the saliency map based on graph theory (GB) [69]. They have formed the activation map based on Itti's feature maps and normalize it using graph theory, but the method is highly computation intensive. Gao et al. have shown saliency detection model based on discriminant centre-surround hypothesis using mutual information both for static images and video sequences [70]. The saliency map is computed by determining the discrimination power of intensity, colour and orientation which are low level features between the centre location and its surrounding.

The frequency domain processing for saliency detection is well exploited by various

Table 2.2: Summary of literature survey related to saliency detection schemes

Year	Authors	Approach	Spatial	Temporal	Ref.
1998	Itti et al.	Centre-surround framework	√	×	[64]
2003	Ma and Zhang	Contrast based using fuzzy region growing	√	×	[67]
2005	Bruce and tsotsos	Maximizing information framework	√	×	[37]
2005	Itti and Baldi	Centre-surround framework	√	√	[71]
2006	Liu and Gleicher	Contrast based using region growing framework	√	×	[68]
2006	Harel et al.	Graph theory	√	×	[69]
2007	Gao et al.	Centre-surround framework	√	×	[70]
2007	Hou and Zhang	Fourier -transform based spectral residual	√	×	[72]
2008	Guo et al.	Phase spectrum of Fourier transform	√	√	[73]
2009	Achanta et al.	Frequency-tuned framework	√	×	[74]
2009	Yu et al.	Pulse discrete cosine transform	√	√	[75]
2009	Cui et al.	Temporal spectral residual	√	√	[76]
2010	Goferman et al.	Context aware saliency	√	×	[77]
2010	Mahadevan et al.	Centre-surround framework	√	√	[78]
2010	Hua at al.	Phase spectrum of Fourier transform	√	√	[79]
2012	Feng et al.	Amplitude spectrum of Fourier transform	√	×	[80]
2013	Imamoglu et al.	Centre-surround framework	√	×	[81]
2014	Lu et al.	Co-occurrence histograms framework	√	×	[82]
2015	Imamoglu et al.	Space-based framework	√	×	[83]

researchers [72–75]. In Fourier transform, the amplitude spectrum represents the magnitude of each frequency component present in the image, whereas the phase spectrum represents the positional information of these frequencies[84]. So, either the amplitude spectrum [72] or the phase spectrum [73] has been chosen as major component to decide saliency in a scene. Hou et al. have proposed a model which is purely computational, Fourier transform based and independent of any biological features [72]. It determines the saliency map with spectral residue (SR) of the log spectrum using the Fourier transform of an input image. However later, Guo et al. have shown that the amplitude spectrum is irrelevant for saliency computation, while the phase spectrum of the Fourier transform (PFT) is only sufficient for the task [73]. But they have calculated saliency only for one scale of resolution (64×64) of image and that leads saliency detection outcomes to be scale dependent.

Achanta et al. have generated the full resolution saliency map unlike Itti [64] and Y.F. Ma [67]. They have used bandpass filter to preserve more frequency content by frequency tuning (FT) compared to other methods and extract features of colour and luminance using difference of Gaussian pyramid [74]. Ying Yu et al. [75] and recently, Hou et al. [85] have proposed pulse DCT based saliency map detection. The pulse DCT is derived from

retaining sign information of the DCT coefficients, which is the phase component of the DCT coefficients. Hence, in pulse DCT, the saliency map is generated by reconstructing the image by applying inverse DCT operation over phase information generating pulse sequence of $+1$ and -1. The pulse sequence resembles firing pulses of neurons and the pulse DCT simulates the iso-suppression properties of similar feature tuned neurons; thereby yielding amplified intensity at a discontinuity. Similar to PFT [73], pulseDCT also resizes the image into $64-$pixels wide to reduce the homogeneity. The real-world objects may be of different shapes and sizes. In addition, the proximities of objects and viewing angle of the viewer or camera may differ and hence the information varies by significant amount over the different scales [86]. If we consider only one fixed scale of input for saliency detection, it may give the limited information in the saliency map. A new type of saliency feature known as context aware saliency is proposed by Goferman et al. They have extracted salient region rather than fixation points from the scene. To determine the salient regions, low level features along with high level factors such as face are used [77]. Recently, Fang et al. have proposed saliency detection method based on amplitude spectrum of quaternion Fourier transform (QFT) [87] and also considering human visual sensitivity for deciding conspicuous location in the scene [80]. Imamoglu et al. have proposed saliency detection scheme using wavelet transform[81]. They have employed wavelet transform to generate multi-scale low level features such as texture and edges. New saliency detection models are proposed recently with different approaches from its predecessors such as 2D co-occurrence histograms based saliency detection [82] and space-based saliency detection [83]. However, these techniques are highly parametric.

Recently, spatio-temporal saliency detection schemes are also gaining popularity. In a video surveillance application, detection of moving objects or capturing surprising events in a complex background scenario have both kinds of motion; interesting and uninteresting. This leads to another dimension of research for saliency detection [71, 88]. Itti and Baldi have proposed modified model of their previous saliency detection model [64] for extracting the surprise or sudden change in information in a dynamic environment [71]. They have included motion saliency and flicker saliency as additional low level features along-with intensity, colour and orientation. And subsequently, a sudden change, also known as a surprise is detected at some pixel locations in every feature map using information theory.

Cui et al. have proposed temporal spectral residual (TSR) based on motion saliency detection for video data [76]. The TSR motion saliency detection method is a modified version of SR saliency detection method proposed by [72]. The Fourier spectrum analysis is done on video data without having any prior information. In addition, motion saliency is calculated using global threshold selection and a saliency majority voting operation on spectral video data. Similarly, Mahadevan and Vasconcelos have also extended their work of spatial domain saliency detection [70] to spatio-temporal saliency detection [78]. In this approach, features are spatio-temporal in nature and discrimination between centre and

surrounding window is measured to determine spatio-temporal saliency. Hua et al. have proposed spatio-temporal saliency in scale-space for tracking and video re-targeting [79]. They have employed phase spectrum of Fourier transform proposed by [73] in coarse to fine search at each time instant.

The summarized literature survey related to saliency detection is given in Table 2.2.

However, many of these saliency detection schemes have restricted use in real-world applications. Some of the difficulties are like highly parametric approach, unbalanced weights for different features, uneven detection of saliency regions such as failing to determine the entire salient region or loosing salient object boundaries or bias towards edges or corners. Even some global statics based saliency detection methods fail for low contrast images.

2.2 Directional Transforms

The recent developments in video acquisition and display systems and exponential growth in transmission bandwidths have increased the demand of superior quality video contents in multimedia applications with resolutions ranging from 176×144 pixels (QCIF) to 3840×2160 pixels (UHD). With widespread adoption of emerging applications like video streaming, video surveillance, blue-ray disk video, etc. video compression has become an integral component of such multimedia applications. However, a video data in an uncompressed format demands a huge amount of storage space and transmission bandwidth. To surpass these physical constraints, an efficient video compression scheme is always required. Various video coding methods have been developed in literature to accomplish video compression such as entropy coding [89], predictive coding [90], block transform coding [6], wavelet/sub-band coding [91]. Block transform coding is the one which is highly exploited in image and video coding by reducing the inherent spatial redundancies between neighbouring pixels. Easy implementation, higher coding gain and unitary transforms are some of the features which have made block transform coding as a prime candidate for video compression systems.

The Karhunen-Loeve transform (KLT) is an optimal transform in block transform based coding as it fully decorrelates the block in the transform domain [92]. However, higher implementation complexity and extra overhead bits are the reasons for the restricted use of KLT in most of the video coding standards. The discrete cosine transform (DCT) is a good approximation of KLT in terms of coding gain. The DCT is widely accepted as an alternative of KLT. Fast implementation techniques, superior compression gain and hardware adaptability are some of the favourable characteristics of DCT that make it most popular transform for video coding [93, 94]. Many image and video compression systems such as JPEG [95], H.26X [6, 13, 16, 17] and MPEG-1/2/4 [14, 15, 20] employ DCT. In H.264/AVC, a multiplication free integer version of 2D-DCT is used for intra-predicted (IP) residuals

(intra-frame coding) or motion-compensated (MC) residuals (inter-frame coding). Recently, high efficiency video coding (HEVC) [21] is proposed to achieve higher compression gain than H.264/AVC. HEVC contains various advanced coding tools such as coding tree block architecture, 33 directional intra-prediction modes, multiple size ($4 \times 4, 8 \times 8, 16 \times 16$ and 32×32) integer transform support and many more [96].

Since the conventional 2D-DCT is a separable transform, it is implemented by applying two 1D-DCTs horizontally and vertically. This characteristics of 2D-DCT makes it a well preferred transform for blocks containing vertical and horizontal directional features. However, the performance of DCT is dubious for other direction-dominant blocks. For diagonal featured blocks, the DCT generates a large number of non-zero coefficients that deteriorates compression gain. Intra-frames of H.264/AVC applies intra-prediction, newer coding tool, to exploit directional correlation among neighbouring pixels in spatial domain [6]. It uses various directional IP-modes (4 modes for 16×16 macroblock and 9 modes for 4×4 or 8×8 block) to mitigate the directional features dominance and improve compression performance. Yet, the IP-residuals show strong directional correlation.

Another area of major concern is coding of MC-residuals in inter-frames. In MC, smooth areas or regions of moving objects with non-translational motions are well predicted due to spatial correlations among neighbouring pixels. Further, the textured backgrounds, object boundaries or edges are high prediction error regions. As in MC-residual frames, these high prediction error regions form 1D-structures with various orientations and encoding of MC-residuals with conventional 2D-DCT lead to lesser compression gain. Therefore, MC-residuals should not be encoded in the same manner as IP-residuals which have 2D-structures. It is proposed by Kamisli and Lim [97] that 1D-DCT would be a better choice to encode such directional 1D-structures rather than conventional 2D-DCT.

In transform based video coding, the coding performance strongly depends on a transform kernel. For a directional block, an appropriately selected transform not only efficiently decorrelates block data, but also improves compression ratio by representing the block with fewer coefficients. As the performance of conventional 2D-DCT for a directional block is sub-optimal, several related works have been reported in literature to find a suitable transform scheme for such directional featured blocks by including directionality into the block transform. There are two approaches to incorporate directionality. The first approach prefers conventional DCT only, but includes some pre-processing operation such as rearrangement of block data to a particular direction [97–105]. For instance, Zeng et al. have proposed directional DCT (DDCT) framework for directional image and video coding [98]. In this framework, eight directional modes are defined, similar to H.264 prediction modes excluding the dc mode. The primary transform selects a particular directional mode and the resultant coefficients are rearranged in such a way that secondary transform exploits the correlation between coefficients. The proposed framework has shown significant improvement in coding performance for directional dominant blocks as compared

Table 2.3: Summary of literature survey related to directional transform

Year	Authors	Approach	Intra-coding	Inter-coding	Ref.
2008	Zeng and Fu	Directional DCT (DDCT)	√	√	[98]
2008	Ye and Karczewicz	Bi-intra prediction and multiple directional transforms	√	×	[106]
2010	Cohen et al.	Direction adaptive transform (DART)	√	√	[99]
2010	Chang et al.	Direction-adaptive partitioned block transform (DA-PBT)	√	√	[100]
2010	Peng et al.	Directional filtering transform	√	×	[101]
2011	Kamisli and Lim	1-D transform for MCP-residuals	×	√	[97]
2012	Yeo et al.	Mode-dependent DCT and DST	√	×	[108]
2012	Wang et al.	Pixel-wise directional intra prediction	√	×	[102]
2012	Han et al.	Jointly optimized spatial prediction and asymmetric DST	√	×	[109]
2012	Gu et al.	Rotated orthogonal transform (ROT)	×	√	[110]
2013	Gabriellini et al.	Adaptive transform skipping	×	√	[103]
2013	Saxena and Fernandes	DCT/DST based transform	√	×	[111]
2013	Cai and Lim	Multiple transform selection	√	√	[112]
2014	Wang et al.	Content adaptive transform framework (CAT)	√	√	[104]
2015	Zhang et al.	All phase bi-orthogonal transform (APBT)	√	×	[105]

to conventional 2D-DCT. However, multiple directional transforms and scanning patterns and extensive use of variable length DCTs are a major bottleneck of this framework. Similarly, Chang et al. have proposed to exploit the directional featured blocks by one of eight directional modes transforms along-with non-directional 2D-DCT [100]. Each directional mode has it own directional transform basis, block partitions and scanning order. In this scheme, 1-D DCT length is limited upto block-size by using block partitioning. However, multiple length 1-D DCTs still exist.

Another approach derives new directional transform kernels by exploiting directional block information or use more than one transforms; a separate transform for each block to improve coding performance [106–111]. A combination of even type-II DCT (EDCT-2) and odd type-III DST (ODST-3) is used according to dominant directional edges in the block [106–109, 111]. However, many of these transform schemes decide the direction of the transforms through training data. But, they require more memory storage for transforms. Multiple transform modes increase the implementation complexity significantly for both encoder and decoder. A further alternative scheme such as pixel-wise directional intra-prediction (PDIP) method utilizes the adjacent reconstructed pixels of different orientation to predict the current pixels keeping the transform module unaltered. It has shown the improvement in bit-rate by 2.5% for intra-coding [102].

As aforementioned, the MC-residuals have different characteristics than image/ intra residuals blocks and applying the same transform to such blocks leads to inefficient compression performance. In [97], Kamisli and Lim have shown that MC-residuals form directional 1D-structures and performing traditional 2D-DCT transform on such blocks would unnecessarily generate more number of coefficients. A set of transforms and

corresponding scanning patterns are used to encode each directional block with most suitable transform mode. But, it increases computational complexity for an encoder. Gu et al. have shown that rotated orthogonal transform (ROT) is a better alternative of DCT for MC-residuals coding [110]. The MC-residual frame is divided into transform region and separate ROT is generated for each transform region by convex function constraints. Since ROT kernel is modified DCT basis by minimizing the orthogonal-constrained L1 norm, it inherits all the merits of DCT's orthogonal kernel and adds directionality in it using rotation matrices.

A suitably designed directional transform yields uncorrelated coefficients and enhances compression gain. Most of the directional transforms available in literature favour rate-distortion optimization (RDO) method for the selection of an optimum directional transform mode [97–99]. RDO is a brute-force approach. It increases encoder complexity by manifold and restricts its use in real-time applications. Recently, Cai at el. [112] have proposed two new algorithms to determine a best suitable transform for each block when multiple transforms are available based on selection of best transform and best number of coefficients to preserve maximum energy compaction. But it is observed that most of these directional transforms face several issues and their practical implementations are restricted in most of the video coding applications. For instance, use of different length DCTs cause 'mean weighting defect' which generates unnecessary non-zero coefficients [97–100]. Further, use of too many directional DCTs also increases number of DC coefficients that require large number of bits to represent by entropy coding as compared to its counter part lower valued AC coefficients. Moreover, introducing of a number of scanning patterns to coincide with the characteristics of each transform mode, leads to higher encoding bit-rate due to extra overhead as side information and increases encoder complexity as well. Finally, our major concern is the computational complexity for selecting an optimum directional transform mode for each directional block. The literature survey related to directional transform is summarized in Table 2.3.

Though we find many directional transforms and associated methodologies in the literature, we strongly feel that there is sufficient scope of future investigation for finding an optimal set of directional transforms that will yield quite high compression ratio.

2.3 Motion Estimation

In video coding, inter-coding exploits temporal redundancy between successive frames and yields superior compression performance. Motion estimation (ME) is an essential tool of inter-coding that determines motions of blocks/ pixels with respect to reference frame/ frames. All video coding standards such as H.261 [13], MPEG-1 [14], MPEG-2 [15], H.263 [17], MPEG-4 part 2 [20],H.264 (MPEG-4 part 10) [6] and high efficiency video coding (HEVC) [21], use ME schemes to exploit temporal redundancy between successive video

frames to achieve higher compression ratio. ME is also used in various video processing applications such as frame interpolation or frame rate up-conversion [113], object tracking and video surveillance [114].

There are various ME methods such as global motion estimation model (GMC) [115], block matching motion estimation algorithm (BMME) [116, 117], phase correlation based ME [118], optical flow [119], pel-recursive approach [120] and parametric-based ME model [121]. Among these methods, the BMME is widely used in most of the video coding standards due to its ease in implementation and repeatability. These properties of BMME are readily exploited in soft-core and VLSI implementation [122, 123]. Technically, BMME is used to determine the displacement of the current block by searching the best matched block in the reference frame/ frames. The displacement is given in terms of motion vector (MV) with respect to the co-located block. The MCP-residuals are the difference between the current block and a reference block, are transformed, quantized and then entropy encoded. Encoding residuals and MVD, the difference between MV and its prediction, significantly reduce the number of bits required to represent a block and MV respectively rather than encoding original content. An efficient ME scheme reduces energy in MCP-residual frames and improves compression ratio. But, ME is computational intensive and consumes almost 60% to 80% of overall encoder complexity in H.264 [124]. Therefore, a fast ME scheme plays an important role in real-time video applications.

There are two types of error modal based BMME schemes: (a) uni-modal error surface based BMME schemes and (b) multi-modal error surface based BMME schemes. These scheme are described below.

2.3.1 Uni-modal error surface based BMME schemes

The uni-modal error surface based BMME schemes are based on following assumptions.

1. For a 2-D translational motion (in x-y plane), pixel intensity remains constant.

2. Error surface is uni-modal i.e. block matching error increases monotonically as the distance from the global minimum increases.

In BMME, full search (FS) also known as exhaustive search [125] is a brute-force approach that yields optimal solution by finding the global minimum. For each block, if the search window of size W_s, then the total number of search points for FS are $(2W_s+1)^2$. As it checks all the search points within the search window, complexity of FS is very high, which restricts its use in real-time video applications. A number of fast BMME algorithms have been proposed in literature that reduce computation complexity significantly as compared to FS and also maintain an acceptable visual quality. These BMME algorithms can be classified into three categories:

1. fixed pattern based BMME (FP-BMME),

2. reduced search points based BMME (RSP-BMME) and

3. lower complexity based BMME (LC-BMME).

In FP-BMME algorithms, fixed patterns are used to check only a few search points within the search window rather than all search points. This includes three-step search (TSS) [126], new three-step search (NTSS) [127], four-step search (FSS) [128], unrestricted centre-biased diamond search (UCBDS) [129], diamond search (DS) [130], hexagon-based search (HEXBS) [131] and octagonal search algorithm[132]. It is observed that these FP-BMME algorithms are easy to implement. But, they may get trapped into a local minimum specifically for fast-motion videos. This results in inaccurate MV leading to high MCP-residuals and eventually degrading compression performance. In RSP-BMME algorithms, the number of search points are limited by successive minimization of the prediction error in a search window. These algorithms exploit spatio-temporal MV correlation of neighbouring blocks of current and/or reference video frames to predict the global minimum using motion vector prediction (MVP). This predicted global minimum search point is considered as initial search centre and true global minimum is searched around its surroundings. The use of MVP greatly reduces the search space due to spatio-temporal correlation of neighbouring blocks. This also reduces the chance of it getting trapped into local minimum. Among RSP-BMME algorithms, few extensively used algorithms are cross-diamond search (CDS) [133], adaptive rood pattern search (ARPS) [134], enhanced predictive zonal search (EPZS) [117], hybrid unsymmetrical-cross multi-hexagon-grid search (UMHexagonS/UMH) [116], diamond and hexagon search (DHS) [135], content-adaptive fast ME [136] and optimized predictive zonal search (OPZS) [137] . The EPZS is an efficient BMME algorithm in terms of computational complexity and compression performance. The EPZS uses various spatio-temporal MVPs and threshold based multiple stopping criteria. It employs simple and efficient diamond and square search patterns to find the global minimum of a block. EPZS yields visual quality as good as FS while significantly reducing the number of search points. UMHexagonS/UMH is another very successful BMME algorithm present in literature. It also uses MVP based initial search centre along with unsymmetrical cross and multi-level hexagon search patterns and threshold based early termination techniques. It is observed that hexagon search pattern checks less number of points compared to diamond search pattern to find same MV. These BMME algorithms (EPZS and UMH) are adopted by H.264 joint model (JM) along with FS due to their superior performances [138]. The LC-BMME algorithms have mainly focused on reducing the computational complexity for each search. Some of the popular methods are based on pixel sub-sampling [139], partial or simple cost estimation [140], histogram [141], successive elimination [142] and multi-layer approaches [143].

Table 2.4: Summary of literature survey related to motion estimation

Year	Authors	Approach	Ref.
		Uni-modal error surface based	
1981	Jain and Jain	Full Search or exahaustive search	[125]
1981	Koga et al.	Three-step search (TSS)	[126]
1994	Li et al.	New three-step search (NTSS)	[127]
1995	Nam et al.	Pixels sub-sampling	[139]
1996	Po and Ma	Four-step search (FSS)	[128]
1996	Lin and Tai	Partial or simple cost estimation	[140]
1998	Than et al.	Unrestricted centre-biased diamond search (UCBDS)	[129]
2000	Zhu and Ma	Diamond search (DS)	[130]
2001	Zhu et al.	Hexagon-based search (HEXBS)	[131]
2002	Cheung and Po	Cross-diamond search (CDS)	[133]
2002	Nie and Ma	Adaptive rood pattern search (ARPS)	[134]
2002	A. M. Tourapis	Enhanced predictive zonal search (EPZS)	[117]
2002	Chen et al.	Hybrid unsymmetrical-cross multi-hexagon-grid search (UMH)	[116]
2009	Cui et al.	Octagonal search	[132]
2009	Cheng etal.	Diamond and hexagon search (DHS)	[135]
2012	Nisar et al.	Content adaptive fast motion estimation	[136]
2012	Park et al.	Histogram	[141]
2013	C.-S. Park	Successive elimination	[142]
2014	Paramkusam and Reddy	Multi-layer approaches	[143]
2015	Abdoli et al.	Optimized Predictive Zonal Search (OPZS)	[137]
		Multi-modal error surface based	
1995	J. Kennedy	Conventional PSO	[144]
1998	Y. Shi	Advanced PSO	[145]
1998	Y. Shi	linearly decreasing weighted PSO (LDWPSO)	[146]
2003	Gong and Ding	Genetic algorithm (GA)	[147]
2004	Ratnaweera et al.	Time-varying acceleration coefficient - PSO (TVAC-PSO)	[148]
2006	Ren et al.	PSO - zero-motion prejudgement (PSO-ZMP)	[149]
2008	Yuan and Shen	Improved PSO (IPSO)	[150]
2012	Erik Cuevas	Harmony search (HS)	[151]
2012	Cai and Pan	Modified time-varying acceleration coefficient variant of PSO	[152]
2013	Cuevas et al.	Artificial bee colony (ABC)	[153]
2013	Pandian et al.	Pattern based BMME scheme based on PSO (PBPSO)	[154]
2014	Fei et al.	Artificial fish-swarm	[155]
2015	Jalloul and Al-Alaoui	Cooperative ME based on multi-swarm PSO	[156]

2.3.2 Multi-modal error surface based BMME schemes

It is observed that all video sequences do not follow uni-modal error surface assumption. In fact, an error surface may have many minima in a block. Therefore, employing uni-modal

error surface based BMME schemes may lead to getting trapped to a local minimum. The popular approach for multi-modal error surface based ME is evolutionary methods such as genetic algorithm (GA) [147], particle swarm optimization (PSO) [152, 154, 156], harmony search (HS) [151], artificial bee colony (ABC) [153] and artificial fish-swarm[155]. In general, the evolutionary algorithms ensure global minimum solution but at the cost of high execution time, since accurate determination of MV is accomplished only after a large number of iterations. PSO is population-based stochastic search technique and is found to be more effective to solve local minima problem [144]. Since the PSO algorithm is designed for global solution, it is not optimized for higher speed. The computational complexity of conventional PSO is very high and hence its use is limited in real-time video encoding. Actually, in conventional PSO, the accuracy of true MV depends not only on the population size, but also on number of iterations [144]. However, in many PSO based encoding schemes, the number of iterations are reduced by enforcing early termination techniques and the accuracy of MV is compromised [152, 154].

In literature, various PSO based BMME schemes are available that improve the encoding time. Ren et al. have proposed particle swarm optimization - zero-motion prejudgement (PSO-ZMP) BMME scheme [149]. The algorithm initially checks the zero motion vector (ZMV) of a block. If a block is static in nature, then no need to perform the remaining search. Otherwise, neighbouring blocks are used to predict the global minimum and PSO is used over predefined fixed search patterns. Yuan and Shen have proposed improved PSO (IPSO) by modifying the conventional PSO for fast BMME by employing centre-biased search pattern and predictive global minimum centre position based on neighbouring blocks MV [150]. Cai and Pan [152] have proposed advanced PSO (APSO) BMME scheme based on a modified PSO approach along-with some stopping strategies. They have modified time-varying acceleration coefficient variant of PSO (TVAC-PSO) [148] for fixed initial positions of particles and close to global minimum based on neighbouring blocks MV. Similarly, a pattern based BMME scheme based on PSO (PBPSO) is proposed by Pandian et. al. [154]. Since the initial particle positions are randomly chosen in conventional PSO, the proposed algorithm uses centre biased fixed diamond or square search pattern with nine particles for initial particle positions. The summarized literature survey related to motion estimation is given in Table 2.4.

Based on extensive literature survey, we observe that there exists a scope for improvement in performance of ME through efficient directional search pattens for uni-modal error surface and PSO based schemes that reduce number of iterations for multi-modal error surface.

2.4 Conclusion

This chapter aims to provide a complete scenario of some existing schemes related to foveated video coding, directional transform and motion estimation. Due to space constraint, only a few important schemes are presented in this chapter. It is observed that the use of these scheme are restricted in real-world applications. Either they do not exhibit any promising results or they are highly parametric or computational intensive. Hence, there is sufficient scope to develop more efficient foveated video compression schemes to improve the compression performance. We hope that improving other tools of video compression scheme such as directional transforms and motion estimation will lead to further improvement in compression performance.

Some efficient foveated video compression schemes are developed in the next chapter.

Chapter 3

Development of Foveated Video Compression Schemes

Preview

In the field of image and video compression, the trade-off between visual quality of a pictorial representation and its compression ratio has been often optimized by exploiting non-uniform sampling property of human visual system (HVS). The important or salient regions are compressed with higher visual quality, while the non-salient regions are compressed with higher compression ratio. To determine the salient regions in a scene, two saliency detection techniques; multi-scale phase spectrum based saliency detection (FTPBSD) and sign-DCT multi-scale pseudo-phase spectrum based saliency detection (SDCTPBSD) are proposed in this chapter. Based on these saliency detection techniques, foveated video compression (FVC) schemes are developed to improve the compression performance further. The proposed FVC schemes are analysed on JM 18.6 of H.264/AVC platform.

The following topics are covered in this chapter:

- Introduction
- Fundamentals of foveated video coding
- Development of saliency detection techniques
- Development of foveated video compression algorithms: FVC-FTPBSD and FVC-SDCTPBSD
- Experimental results and discussion
- Conclusion

3.1 Introduction

Recently, foveated imaging based image or video compression schemes are in high demand since they not only match with the perception of human visual system (HVS), but also yield

higher compression ratio. The foveated imaging selects interesting regions in each frame and encodes them in priority basis. In the retina of our human eye, only a small region of $2^\circ - 5^\circ$ of visual angle (the fovea) around the centre of gaze is captured with high spatial and colour resolutions, with the resolution falling off logarithmically towards corner end of view due to non-uniform distribution of photoreceptors [157, 158]. Saliency detection is a technique to determine the visually significant regions within a scene, which are different from their surroundings. The feature elements like contrast, colour and orientation are used to specify the relative importance of various regions in a scene [159, 160]. Thus, in principle it may not be necessary or useful to encode each video frame with uniform visual quality [45]. Foveated imaging is a form of lossy compression. The higher the roll off in resolution from the direction of gaze, the greater will be the compression. A conceptual diagram of the proposed foveated video compression (FVC) scheme is shown in Figure 3.1.

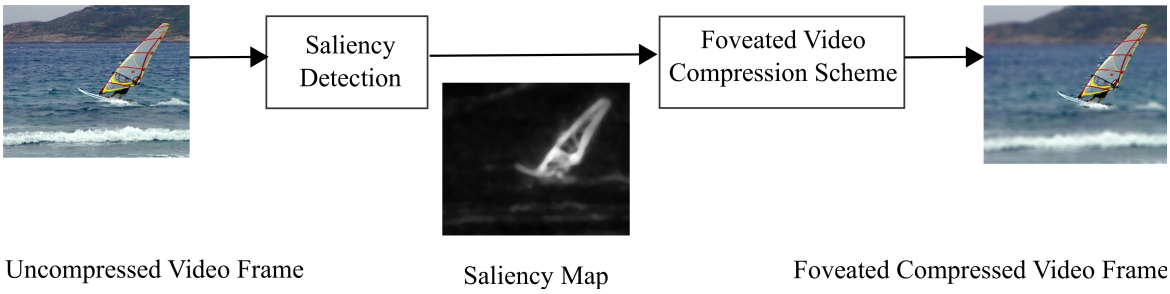


Figure 3.1: Conceptual diagram of the proposed foveated video compression scheme

Based on these observations, we propose two multi-scale saliency detection techniques: (1) multi-scale phase spectrum based saliency detection (FTPBSD) and (2) sign-DCT multi-scale pseudo-phase spectrum based saliency detection (SDCTPBSD). The novel contribution of the proposed FTPBSD technique is determination of saliency map with multi-scale analysis employing phase spectrum obtained from Fourier transform. On the other hand, the proposed SDCTPBSD technique adopts sign-DCT (SDCT) to extract feature maps from multi-scale images. The FTPBSD is a spatial saliency detection technique, whereas SDCTPBSD is a spatio-temporal saliency detection technique. In addition, we have also investigated various fusion methods to combine multi-scale saliency maps for optimum objective performance.

Finally, foveated video compression (FVC) schemes based on these saliency detection techniques are proposed. An object map (binary image) is obtained by threshold based segmentation of a saliency map for each video frame. In subsequent stages, a foveated video is encoded by spatially varying the resolution of a frame based on the Euclidean distance between fixation (salient) and non-fixation points. The proposed foveated video schemes are compared against the conventional non-foveated video coding for H.264/AVC platform.

(a) (b)

Figure 3.2: Example of foveated imaging for *News* sequence : (a) original frame, (b) reconstructed foveated frame

3.2 Fundamentals of FVC and Saliency Map

A conventional video coding scheme encodes video data with uniform resolution, whereas an FVC scheme yields higher compression ratio by varying the resolution of video data similar to fall-off resolution of HVS [43, 45, 50, 53, 58, 59]. There are a number of industry related applications of foveated imaging such as image watermarking [161], scalable video coding [60], video coding [48], advertisement evaluation [162] and 3-D object recognition [163]. An example of foveated imaging is shown in Figure 3.2 for *News* sequence. In Figure 3.2(a) an original frame is shown and Figure 3.2(b) is the reconstructed foveated frame. It can be observed that in the reconstructed foveated *News* frame, visual quality is high at fixation point that is at the dancing girl in the background and visual quality degrades rapidly moving away from the fixation point towards corners of the frame.

In an FVC scheme, visual attention is determined by tracking the eye positions in real-time that yield fixation points or foveated points in a scene. The hardware based eye tracking devices are commercially available that record movement of the eyes. Major drawbacks of these devices are higher cost and inconvenient designs which give strain to the eyes. Therefore, determination of a visual saliency map is considered as a prospective candidate for foveated imaging.

In the emerging field of computer vision, saliency detection has become one of the keen areas of intense research. A salient region is the most important region in a visual scene. The primates move their fovea towards salient object or regions to get highest resolution to these objects. In the field of image and video compression, a good trade-off can be achieved between the visual quality and compression ratio by exploiting the properties of HVS [60]. According to Koch and Ullman, conspicuous locations of a visual scene is guided by the intensity level of activity of receptors for different features (such as intensity, colour, orientations and direction of movements) and thus feature maps are extracted at pre-attentive

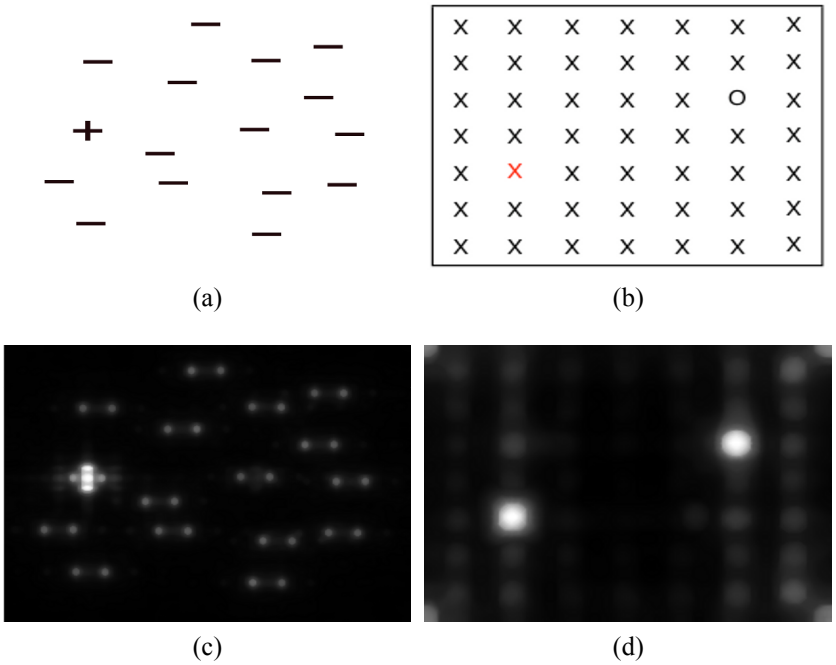


Figure 3.3: Examples of saliency map: (a) Input pattern with dash and plus, (b) Input pattern with cross and circle, (c)-(d) Saliency maps of (a) and (b), respectively

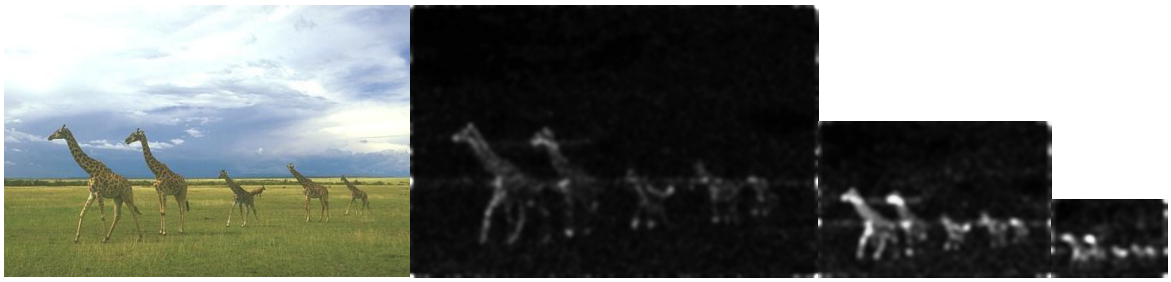
stage [164]. Each feature map registers individual conspicuous locations in primary visual cortex [165]. At later stage, these separable feature maps are combined to generate a global conspicuous location map known as 'saliency map' or 'activation map' for the visual scene. Thus, a saliency map represents the prominence of each and every location of a visual scene [157, 166].

Some examples of saliency maps are shown in Figure 3.3. In Figure 3.3(a), a single *plus* sign is surrounded by many *dash* signs. Since *plus* sign is different from its surrounding in shape, it is observed that *plus* sign pops out in saliency map as shown in Figure 3.3(c). Similarly, in Figure 3.3(b) red coloured *cross* and black coloured *circle*, surrounded by black coloured *cross*, are different from their surrounding in colour and shape, hence they pop out as salient objects as shown in Figure 3.3(d).

3.3 Development of Saliency Detection Techniques

Saliency detection is a technique which predicts significantly important regions in a scene. Despite much research being done in the field of saliency detection, it is still a challenging task to achieve high accuracy in detecting salient objects in real-time. It is understood that an efficient saliency detection scheme should have the following properties [74].

- Scale-space invariance;
- High discrimination capability to detect salient regions;
- Low computational complexity for its suitability in real-time applications;



(a) (b) (c) (d)

Figure 3.4: Example of multi-scale saliency maps: (a) Original image, (b)-(d) represent saliency maps at different level of Gaussian pyramid ranging from level 0 to level 2

- Independence from tuning parameters and *a priori* information;
- Uniform emphasis on the whole salient areas and capability to maintain well-defined salient object boundaries.

In a real world, a scene contains various objects of different shapes and sizes. It is possible that the proximities of objects may differ from the direction of a person's viewing angle. These irregularities between objects, in distances, orientations and viewer's gaze angles, are the major bottlenecks for developing an algorithm to extract features for vision analysis, because analysis of one scale may not have information at another scale or resolution [167]. For instance, as shown in Figure 3.4, in level 0 which is base level, all giraffes are appearing as salient objects. However, in saliency maps of coarser levels, the first two giraffes have higher degree of saliency. Overall, the second giraffe region represents highest prominence. Hence, to achieve scale invariant properties and accurate measurement of saliency maps, it is proposed to have multi-scale salient object analysis.

For an input image f of size $H \times W$ pixels, output saliency map SM is represented as:

$$SM = \mathbb{R} : [0, 1] \tag{3.1}$$

where $SM(i, j)$ indicates level of conspicuousness for a pixel at spatial-coordinate (i, j) of a scene. Salient region pixels will have higher values as compared to non salient regions.

The proposed schemes take an input f, a colour image of dimension $H \times W \times C_k$, where H,W and C_k represents number of rows, columns and colour channels of image respectively. If the image is RGB then $C_k = 3$, while for gray image $C_k = 1$. If the input is a video data, f will be a video frame at time instant 't_k'. The input image f is initially converted to CIE L*a*b* colour space [168]. The CIE L*a*b* colour space is based on colour opponent characteristics of a human visual field, so it shows perceptually uniform colour distribution and component L closely resembles human perception of intensity [4]. The input image f is subjected to a Gaussian filter (g) to blur the image so that unwanted noise will be removed. The blurred image $\bar{f}$ is obtained as:

$$\bar{f}(i,j,c_k) = \sum_{k=-1}^{1}\sum_{l=-1}^{1} g(k,l) f(i+k, j+l, c_k) \tag{3.2}$$

where i and j represent image coordinates as $i = 0, 1, 2, \ldots, H-1$, $j = 0, 1, 2, \ldots, W-1$ respectively and c_k depicts colour channels.

To generate multi-scale structure, an image is decomposed by lowpass filtering in the form of pyramids which is known as Gaussian pyramid. The Gaussian pyramid is obtained by smoothing or blurring the image by a Gaussian smoothing filter 'kernel' to overcome the aliasing effect and subsequently sub-sampling the smoothed image by a factor of half in both horizontal and vertical directions. The same process iteratively repeats for successive levels [62]. For multi-scale analysis, the number of levels (N_L) of pyramid structure is a very crucial parameter. If N_L is very high, the computational complexity and storage space increases exponentially with N_L. However, if N_L is very low, it fails to yield scale invariant feature. The N_L levels multi-scale images ($\bar{f}_0, \ldots, \bar{f}_{N_L-1}$) from coarse to fine levels are obtained by convolving $\bar{f}$ with Gaussian kernel h as [62]:

$$\bar{f}_0 = \bar{f}(i,j,c_k), \qquad \text{base level } n = 0 \tag{3.3}$$

$$\bar{f}_n = \sum_{k=-2}^{2}\sum_{l=-2}^{2} h(k,l)\bar{f}_{n-1}(2i+k, 2j+l, c_k), \quad 1 \le n \le N_L - 1 \tag{3.4}$$

where h is 5-tap filter defined as:

$$h = \left[\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16}\right] \tag{3.5}$$

The spatial saliency maps of L*, a* and b* channels are denoted as $\bar{S}^L, \bar{S}^a$ and $\bar{S}^b$ respectively. Nearly all multi-scale saliency detection methods, present in literature, generate final saliency map by taking average of all saliency maps generated at each scale. The same fusion operation is repeatedly opted on saliency maps generated by different features or channels; for instance intensity and colours. In this chapter, various unification methods are also investigated for combining inter-scale saliency maps, as well as for channel saliency maps, so that an optimal fusion method can be selected for saliency detection.

Both the proposed schemes (FTPBSD and SDCTPBSD) are bottom-up, scale-invariant saliency detection techniques that automatically detect salient objects in a scene and essentially require no prior knowledge of visual stimuli. A spatial saliency map is a unification of saliency maps generated using low level features (such as intensity and colour). On the other hand, a spatio-temporal saliency map is a weighted sum of spatial saliency map and a motion saliency map computed by motion detection low level feature. The proposed saliency detection schemes (FTPBSD and SDCTPBSD) are discussed in details in the following sections.

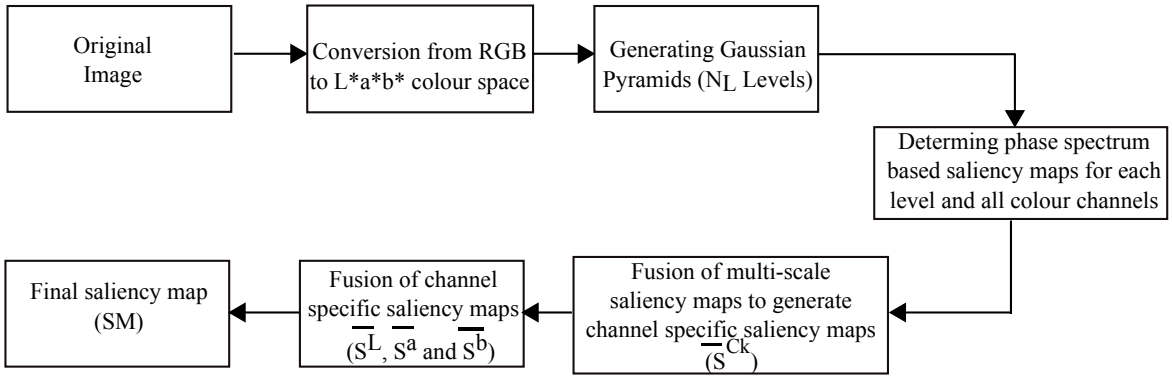


Figure 3.5: Flowchart of proposed FTPBSD method

3.3.1 Multi-scale phase spectrum based saliency detection (FTPBSD)

In this section, we propose multi-scale phase spectrum based saliency detection scheme. In this scheme, the foremost step is to build a Gaussian pyramid structure of input image for N_L levels. In the proposed scheme, we have set $N_L = 3$. In subsequent steps, saliency maps of Gaussian filtered images are computed at each level using Fourier transform. Finally, master saliency map is determined by fusion of all interim saliency maps considering all levels of Gaussian pyramid and for individual colour channels. The schematic representation of our proposed scheme is shown in Figure 3.5.

Determination of Saliency Map

In the proposed scheme, a saliency map is determined by phase spectrum of Fourier transform. Fourier transform converts the spatial domain input image into frequency domain. An image in frequency domain can be expressed as amplitude spectrum and phase spectrum. The two dimensional discrete Fourier transform (DFT) of an image $f(i,j)$ of resolution $H \times W$ pixels is obtained as:

$$F(u,v) = \frac{1}{W \times H} \sum_{i=0}^{H-1} \sum_{i=0}^{W-1} f(i,j) e^{-j_m 2\pi (\frac{ui}{W} + \frac{vj}{H})} \tag{3.6}$$

where $u = 0, 1, 2, 3, \ldots, H-1, v = 0, 1, 2, 3, \ldots, W-1$ and $j_m = \sqrt{-1}$.

Since $F(u,v)$, in general, is complex-valued, it is represented by:

$$F(u,v) = \Re(F(u,v)) + j_m \Im(F(u,v)) \tag{3.7}$$

where $\Re(F(u,v))$ and $\Im(F(u,v))$ represent the real and imaginary parts, respectively.

The Fourier amplitude spectrum $|F(u,v)|$ and phase spectrum $\theta(u,v)$ are defined as:

$$|F(u,v)| = \sqrt{\{\Re(F(u,v))\}^2 + \{\Im(F(u,v))\}^2} \tag{3.8}$$

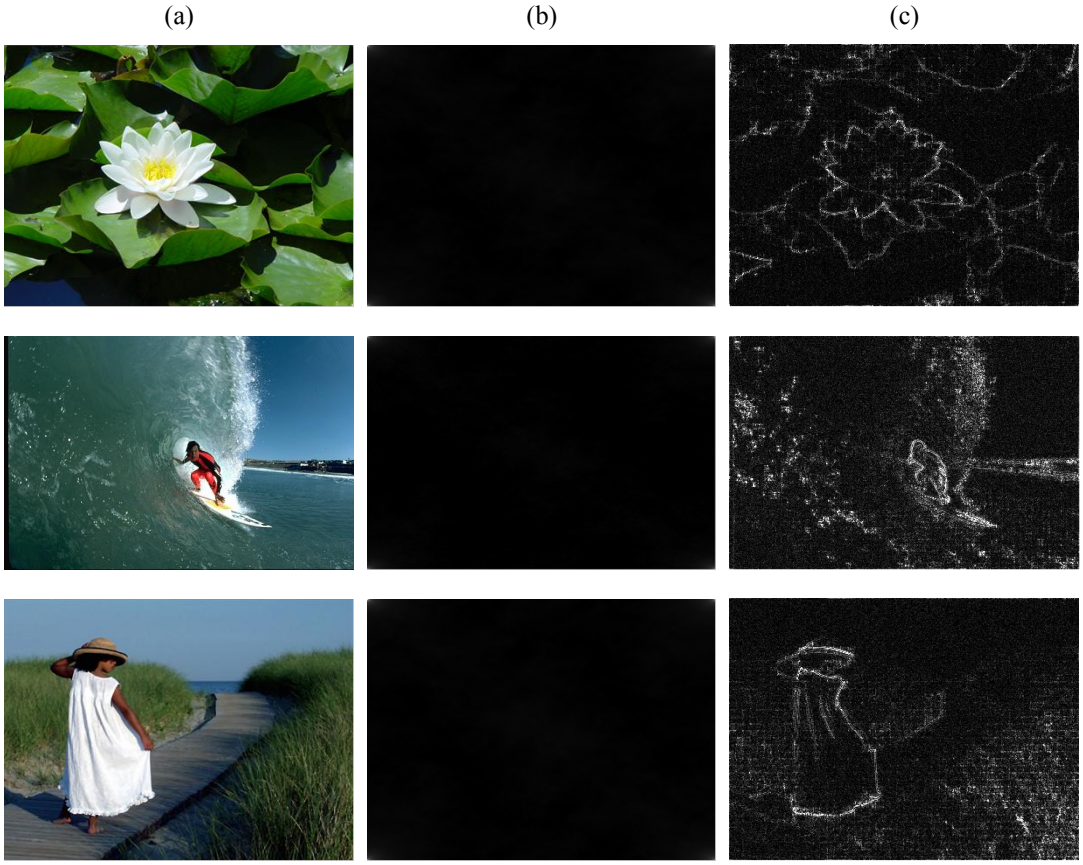


Figure 3.6: Examples of reconstructed images after performing inverse Fourier transform operations on amplitude and phase spectrum individually: (a) Input original image, (b) Reconstructed with amplitude spectrum and (c) Reconstructed with phase spectrum

$$\theta(u,v) = \arctan\left[\frac{\Im(F(u,v))}{\Re(F(u,v))}\right] \tag{3.9}$$

$F(u,v)$ can also be represented in polar form as:

$$F(u,v) = |F(u,v)|e^{j_m\theta(u,v)} \tag{3.10}$$

An amplitude spectrum represents the magnitude of each frequency component present in an image, while phase spectrum shows where these frequency components are present. The positional information is contained in the phase spectrum [84, 169]. Therefore, if we reconstruct an image $\tilde{f}(i,j)$, by taking inverse Fourier transform using phase spectrum $e^{j_m\theta(u,v)}$ alone ($|F(u,v)| = 1$), the homogeneous or similar regions get suppressed as there is no change in phase while non-homogeneous regions different from their surroundings will pop out automatically and yield saliency map of the input image [73]. In Figure 3.6, some of examples are shown for comparison of reconstructed images using only amplitude spectrum and only phase spectrum, respectively.

Therefore, to determine saliency map (S_n), Fourier transform is applied to all levels (N_L) of images $(\bar{f}_n(i,j,c_k))$ individually, where n varies from 0 to $N_L - 1$. For n^{th} level, we have:

$$\bar{F}_n = \mathscr{F}(\bar{f}_n(i,j,c_k)) \tag{3.11}$$

where $\mathscr{F}$ represents the Fourier transform.

Now, inverse Fourier transform is applied to phase spectrum $e^{j_m\theta(u,v)}$ keeping the amplitude spectrum as unity. Subsequently, Gaussian filter (g) is applied before generating saliency map (S_n) at each level. Gaussian filter is employed to diminish the effect of scattered salient pixels and to smoothen the salient regions. The mathematical realization is as follows:

$$S_n(i,j,c_k) = g(i,j) * \left(\mathscr{F}^{-1}\left[e^{j_m\theta(u,v)}\right]\right)^2 \tag{3.12}$$

where $\mathscr{F}^{-1}$ represents the inverse Fourier transform.

Generation of master saliency map and fusion algorithm

In the previous section, we have determined saliency maps for each colour channel comprised of all levels $S_n = S_0, S_1, \ldots S_{N_L-1}$ where $S_n \in \mathbb{R}$. To generate interim saliency map for each colour channels ($\bar{S}^L, \bar{S}^a$ and $\bar{S}^b$), each saliency map (S_n) is up-sampled to the size of input image (f) using bi-cubic interpolation and combined together by applying fusion techniques. Finally, master saliency maps SM is computed by unifying all channels saliency maps.

In this chapter, we have also investigated various fusion method to determine an optimum method which will yield higher objective performance. We have analysed four different fusion methods to combine multi-scale saliency maps as well as colour channels saliency maps. The most widely used and simple method is *averaging*. The drawback of averaging is degree of saliency at particular pixel coordinates in one level may reduce due to co-located non-salient pixel in saliency map of another level. Therefore, overall contrast of salient regions get subdued. *Maximum selection* preserves salient regions and neglects others, which makes it more suitable for the task of saliency detection, but it has a tendency to incline towards edges which may not be salient regions in final saliency map. Two other advanced methods, *local maximum selection* and *local maximum variance* were also considered for fusion rules. Hence, following four fusion techniques are investigated:

- *averaging* takes average of all components at every locations,
- *maximum selection* selects the component which has higher degree of saliency,
- *local maximum selection* keeps the component that has largest sum of the absolute value in a window and
- *local maximum variance* selects the component that has largest local variance around a window.

Algorithm 3.1 Multi-scale phase spectrum based saliency detection (FTPBSD)

Input: Image/Video frame, f of dimensions $H \times W \times C_k$
Output: SM saliency map of dimensions $H \times W$
Method:
1. Input RGB image, $f(i, j, c_k)$.
2. Convert colour space from RGB to L*a*b*.
3. Apply Gaussian filter to blur the image and obtain $\bar{f}(i, j, c_k)$.
4. Define number of levels (N_L) for Gaussian pyramid structure. Set $N_L = 3$.
5. Generate Gaussian pyramid structure $\bar{f}_n(i, j, c_k)$ for $0 \leq n \leq N_L - 1$.
6. Compute saliency map $S_n(i, j, c_k)$ by phase spectrum of 2D-FFT to each level of images for each colour channel using (3.12).
7. Calculate interim saliency maps by applying image fusion algorithm to all saliency maps of each level after up-sampling to size of f:
$\bar{S}^{c_k} = Image\_Fusion(S_0, \cdots, S_{N_L-1})$
8. Determine final master saliency map (SM) by applying image fusion algorithm to all colour channel saliency maps.
$SM = Image\_Fusion(\bar{S}^L, \bar{S}^a, \bar{S}^b)$
9. Normalize SM to $[0, 1]$ by min-max normalization using (3.13).

These fusion methods are applied to all multi-scale saliency maps and to three colour channel saliency maps chronologically. So there are $4^2 = 16$ combinations of fusion methods for the generation of final saliency map. Out of these 16 methods, the method, which demonstrates significantly higher objective performance, is considered for our proposed method. A detailed comparative analysis of best eight combination of fusion methods is discussed in experimental results.

Finally, output saliency map SM is linearly normalized to $[0, 1]$ by min-max normalization as:

$$SM(i, j) = \frac{SM(i, j) - min(SM)}{max(SM) - min(SM)} \tag{3.13}$$

The algorithm for the proposed FTPBSD scheme is presented as **Algorithm 3.1**.

3.3.2 Sign-DCT multi-scale pseudo-phase spectrum based saliency detection (SDCTPBSD)

In principle, the proposed SDCTPBSD scheme uses sign information of discrete cosine transform (DCT) also known as sign-DCT (SDCT) [170]. It resembles the response of receptive field neurons of HVS. The SDCT is applied over multi-scale Gaussian pyramid of an input image to determine the spatial saliency. The final spatial saliency map is generated by enforcing biological approach over intensity and colour feature channels. To determine temporal saliency map, motion of objects between frames is detected using displaced frame differencing method. Subsequently, SDCT is employed to motion data to extract salient regions similar to spatial saliency detection approach. Finally, the bottom-up

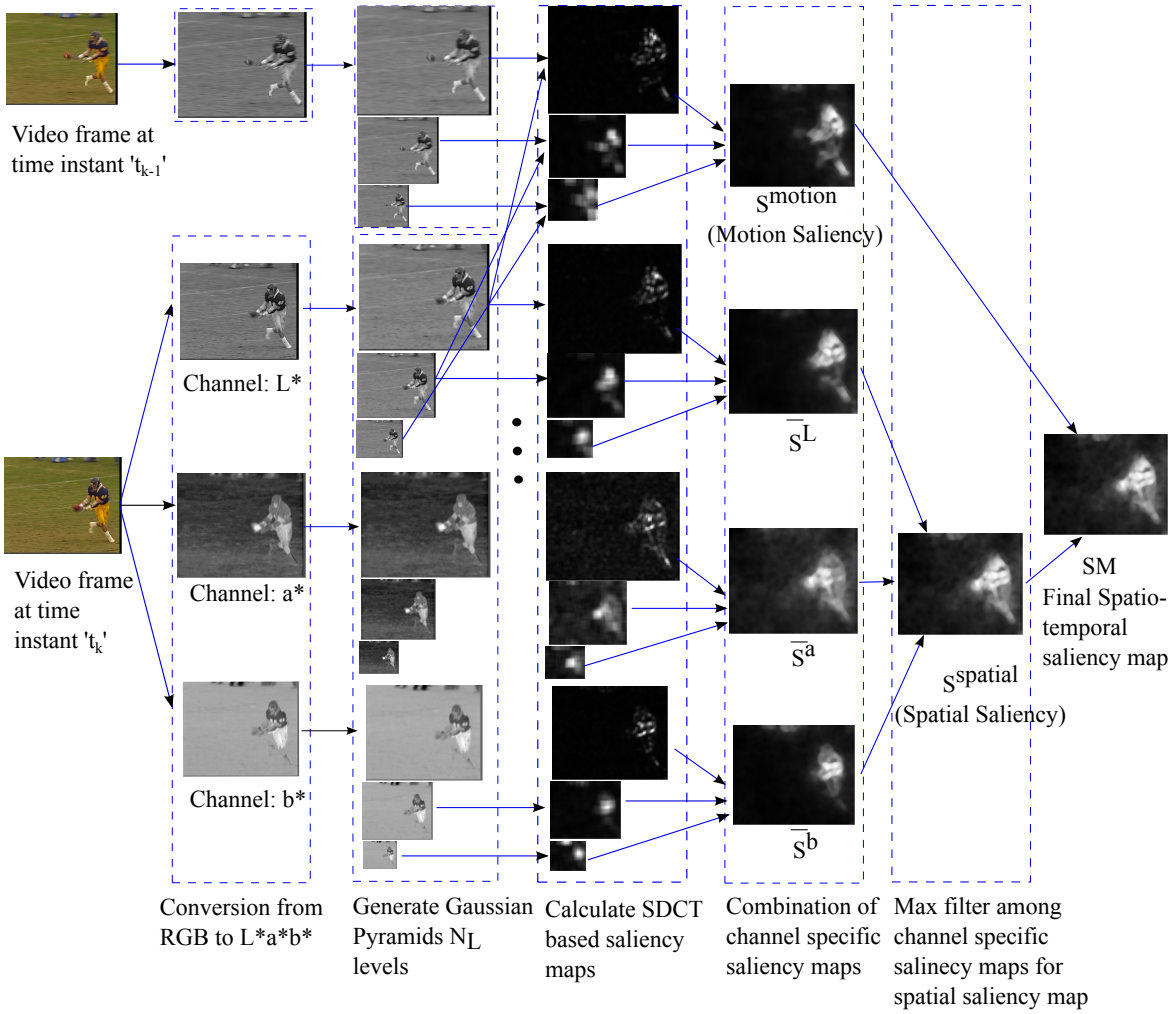


Figure 3.7: Illustration of the proposed SDCTPBSD scheme

spatio-temporal saliency map is obtained by linear weighted sum of these two saliency maps. An illustration of the SDCTPBSD scheme is given in Figure 3.7.

In proposed SDCTPBSD, N_L number of levels of multi-scale images ($\bar{f}_0, \ldots, \bar{f}_{N_L-1}$) are produced. In any multi-scale analysis, determination of number of levels for Gaussian pyramidal structure is the toughest task. In the proposed SDCTPBSD scheme, an optimum number of levels N_L is adaptively calculated based on number of rows (H) and columns (W) of the input image. The N_L is calculated as:

$$N_L = \left\lceil \frac{Log_2(max(W,H)+1}{2} \right\rceil \tag{3.14}$$

where $\lceil\ \rceil$ is the *ceil* operator.

To validate the value of N_L, average F-measures are calculated for images of different resolutions, shown in Figure 3.8. Three groups of different dimensions of same images ($300 \times 400, 150 \times 200$ and 75×100), obtained by sub-sampling original images of 300×400 dimension, are chosen for saliency detection by the proposed scheme and F-measure values are compared for different numbers of levels. It is observed from Figure 3.8 that

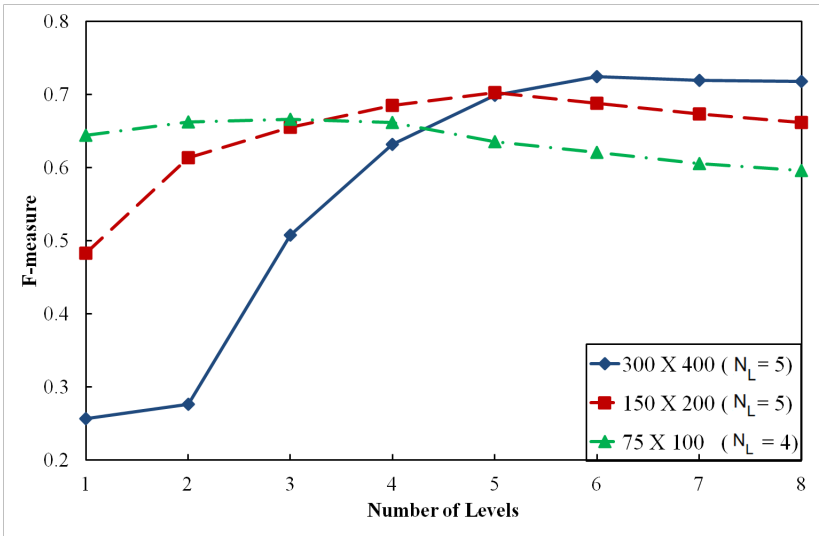


Figure 3.8: Variation in F-measure for different numbers of levels selected for Gaussian pyramid architecture with respect to different resolutions of images. The calculated number of levels (N_L) by the proposed scheme for each resolution images are shown in brackets

F-measures have higher values close to the number of levels those are calculated by N_L for each dimension individually.

The algorithm for the proposed SDPBSD scheme is presented as **Algorithm 3.2**. Saliency maps are computed on multi-scale images for all channels for each scale denotes as $\bar{S}_n^L, \bar{S}_n^a$ and $\bar{S}_n^b$ for L^*, a^* and b^* channels, respectively. For saliency detection, SDCT is computed by initially applying DCT to all multi-scale images of the input image and then extract binary valued $+1$ and -1 sign information of DCT coefficients. These pulses are SDCT coefficients and represent firing responses of neurons for visual stimuli. The proposed scheme simulates the saliency detection functionality of HVS, as saliency is the response of intra-cortical iso-feature suppression activities of primary visual cortex neurons [171]. An illustration of the proposed saliency detection phenomenon for 1-D signal is shown in Figure 3.9 for better comprehension.

To compute motion saliency (S^{motion}) for a video input, the proposed scheme is employed to difference frame which is computed by taking difference between of L^* channels of current video frame at time instant t_k and previous video frame sampled at time instant t_{k-1} at multiple scales. Hence, there are $4N_L$ number of saliency maps available collectively from L^*, a^*, b^* and motion channels. It is proposed to use harmonic mean for combining these multi-scale saliency maps for each individual channels. The objective of this combination process is to reduce the impact of extreme outliers in both ends (high and low). The harmonic mean works well for skewed distribution of data as compared to arithmetic mean or geometric mean. Therefore, a pixel will get privilege for positioning on saliency map only when it is salient in all scales. However, the proposed scheme is also evaluated and compared for arithmetic and geometric mean against harmonic mean outcomes as shown in Table 3.2.

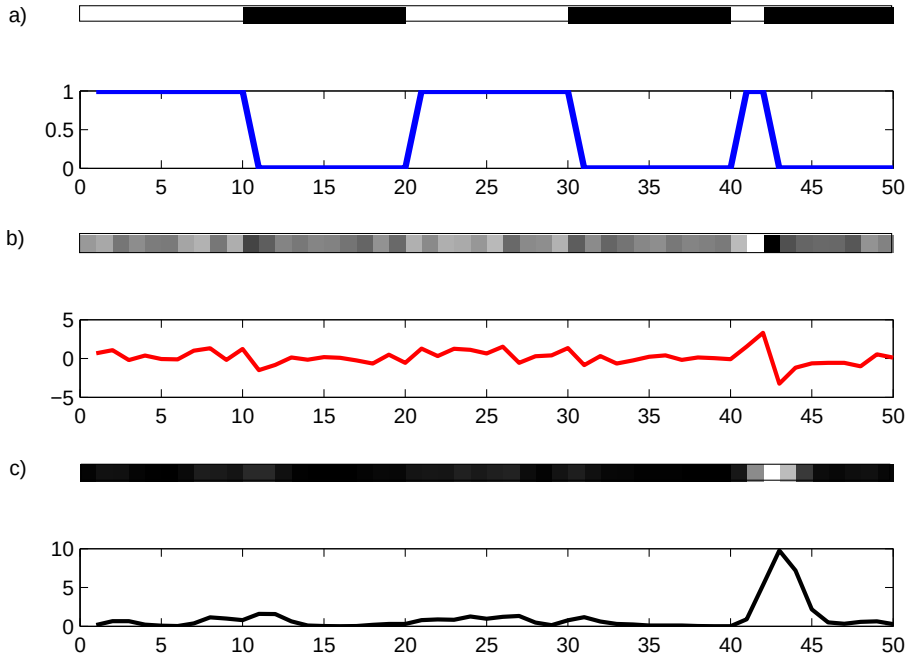


Figure 3.9: Step by step illustration of SDCT based saliency detection on 1-D signal. a) original image representation of 1-D input signal $f(t)$ and its line plot shown below (in all plots x-axis represents samples and y axis depicts amplitude), signal shows periodicity for first two cycles but discontinuity occurred at third cycle; b) image representation and signal plot after applying SDCT and subsequently IDCT over SDCT coefficients, c) final saliency map image representation and signal plot

A median filtering is performed on the resulting four saliency maps ($\bar{S}^L, \bar{S}^a$ $\bar{S}^b$ and S^{motion}) to remove stand alone pixel noise and subsequently, output saliency maps are normalized to the interval $[0, 1]$ by min-max normalization. As the saliency depends on the responses of most active cells of different feature-tuned cells of V1 and there is no discrimination for any preferred feature [172], max filter is applied at every location of $\bar{S}^L$, $\bar{S}^a$ and $\bar{S}^b$ to generate spatial saliency map $S^{spatial}$.

Finally, a spatio-temporal saliency map SM of the proposed scheme is obtained by taking weighted addition of the spatial saliency map $S^{spatial}$ and S^{motion}. The weighting factor $\omega \in [0, 1]$ balances the impact of these two saliency maps on final spatio-temporal saliency map. For instance, $\omega > 0.5$ will give more weight to spatial saliency and $\omega < 0.5$ will emphasize the motion saliency map, whereas $\omega = 0.5$ will provide perfect balance between these two maps. However, it is observed that motion feature has more impact on selective visual attention than the other low level features such as contrast, colour, etc. [173]. Hence, in the proposed method ω value is set to 0.3 to give more emphasis to motion saliency. Finally, spatio-temporal saliency map SM is normalized to $[0, 1]$ using min-max normalization.

Algorithm 3.2 Sign-DCT multi-scale pseudo-phase spectrum based saliency detection (SDCTPBSD)

Input: Video frame, f of dimensions $H \times W \times C_k$
Output: SM spatio-temporal saliency map of dimensions $H \times W$
Method:
1. Input RGB image/ frame, $f(i, j, c_k)$.
2. Convert colour space from RGB to L*a*b*.
3. Apply Gaussian filter to blur the image/ frame and obtain $\bar{f}(i, j, c_k)$.
4. Compute difference frame by L^* channel $\overline{diff_L} = \bar{f}_t(i, j, L^*) - \bar{f}_{t-1}(i, j, L^*)$
5. Define number of levels (N_L) for Gaussian pyramid structure using (3.14).
6. Generate Gaussian pyramid structure $\bar{f}_n(i, j, c_k)$ and $\overline{diff_L}$ for $0 \leq n \leq N_L - 1$.
7. Compute saliency map $S_n(i, j, c_k)$ and S_n^{motion} by applying SDCT to each level of images for each channel including motion channel.

$$A_n(u, v) = sign(DCT(\bar{f}_n(i, j))$$

$$S_n(i, j, c_k) = g(i, j) * \left(IDCT\Big[A_n u, v\Big] \right)^2$$

8. Calculate interim saliency maps by applying image fusion algorithm to all saliency maps of each level after up-sampling by bi-cubic interpolation to size of f:

$$\bar{S}^{c_k} = Image\_Fusion(S_0, \cdots, S_{N_L - 1})$$

9. Employ median filter to interim saliency maps ($\bar{S}^L, \bar{S}^a, \bar{S}^b$ and S^{motion}) to remove standalone pixel noise and normalize it to $[0, 1]$ by min-max normalization.
10. Determine final spatial saliency map ($S^{spatial}$) by applying max filter at every location of all color channel saliency maps.

$$S^{spatial} = max(\bar{S}^L, \bar{S}^a, \bar{S}^b)$$

11. Evaluate the spatio-temporal saliency map.

$$SM = \omega \times S^{spatial} + (1 - \omega) \times S^{motion} \quad \omega \text{ set to } 0.3$$

12. Normalize SM to $[0, 1]$

3.4 Development of Foveated Video Compression Algorithms: FVC-FTPBSD and FVC-SDCTPBSD

The foveated video compression (FVC) scheme achieves higher compression ratio by spatially varying the resolution of a video data. The fixation (salient) points have the highest resolution and there is steep roll-off away from the fixation points. The FTPBSD and SDCTPBSD schemes produce saliency maps which represent the prominence of each pixel in a frame based on intensity, colour and movement of the objects. Technically, the FVC scheme yields a compressed video bit-stream where the salient regions have higher visual quality as compared to non-salient regions. In fact, the visual quality of the non-salient regions degrade exponentially with proximities from salient regions.

The proposed FVC scheme in H.264/AVC platform is shown in Figure 3.10. The algorithm for the proposed FVC scheme is presented as **Algorithm 3.3**. The detailed implementation process is given below.

Step 1:

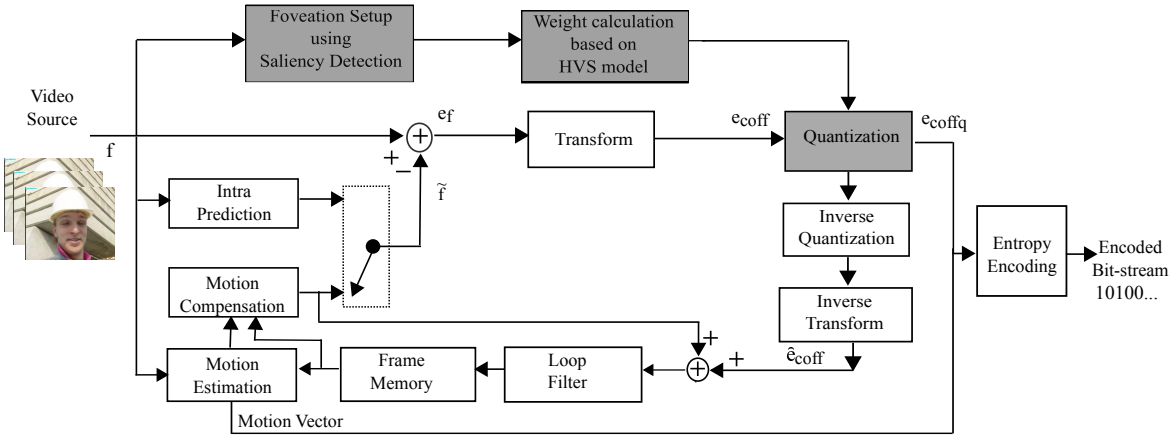


Figure 3.10: Block diagram of Foveated video compression scheme in H.264/AVC platform

Algorithm 3.3 Foveated video compression scheme in H.264/AVC platform

Input: Video frame, f of dimensions $H \times W \times C_k$
Output: Foveated video encoded output
Method:
1. Determine fixation points for a given video frame of size $H \times W$ pixels using one of the proposed saliency detection techniques (FTPBSD or SDCTPBSD).
2. Generate binary object map O by thresholding using fuzzy c-means clustering.
3. Evaluate Euclidean distance, $D_E(i,j)$ between the current pixel $f(i,j)$ location to nearest non-zero pixel $f(k,l)$ location using (3.15).
4. Normalize $D_E(i,j)$ to $[0,6]$.
5. Determine a new foveated QP value (QP_{fov}) for a macroblock using (3.16).
6. Encode the current macroblock with QP_{fov}.
7. Repeat the process for each macroblock in a frame.

Firstly, the fixation points are determined for a given video frame of size $H \times W$ pixels using one of the proposed saliency detection schemes (FTPBSD and SDCTPBSD). The generated saliency map is converted to a binary object map O by thresholding using fuzzy c-means clustering. In the object map O, the salient points are represented as 1 and non-salient points as 0.

Step 2:

Since visual quality will fall exponentially away from fixation points, distances of each pixel from the fixation points are determined. The Euclidean distance, $D_E(i,j)$ is calculated between the current pixel $f(i,j)$ location to nearest non-zero pixel $f(k,l)$ location. The $D_E(i,j)$ is mathematical expressed as:

$$D_E(i,j) = \sqrt{(i-k)^2 + (j-l)^2} \tag{3.15}$$

Step 3:

Since H.264/AVC is a block based encoding scheme, the quantization parameter

(QP) may change at macroblock (16×16 pixels) level. Hence, an average value of D_E for all 16×16 locations are considered to determine a new foveated QP value (QP_{fov}) for a macroblock. In a macroblock, if all points are fixation points, then it will have lower D_E value. Hence, the macroblock will be encoded with minimum QP_{fov} value and will yield higher visual quality. On the other hand, the QP_{fov} will increase exponentially with increase in D_E value and will yield higher compression ratio for non salient regions. Since H.264/AVC has only 52 values of QP, the $D_E(i,j)$ is normalized to $[0,6]$, so that QP_{fov} will not saturate for smaller initial QP value and QP_{fov} will not have sudden huge change which may exhibit blocking artefacts in reconstructed video frames. The QP_{fov} is mathematically calculated as:

$$QP_{fov} = QP + e^{(D_E/2)} \tag{3.16}$$

An example for FVC scheme is shown in Figure 3.11 for *Soccer* sequence. Figure 3.11(a) is the original 002 frame, Figure 3.11(b) is the saliency map generated using SDCTPBSD scheme, Figure 3.11(c) is the object map, Figure 3.11(d) is the normalize D_E map, Figure 3.11(e) is the distribution of QP_{fov}, where QP_{fov} is ranging from 26 to 38 values depending upon the D_E calculated for each macroblock and Figure 3.11(f) is the reconstructed output foveated video frame of encoded original frame with QP_{fov}. It can be observed from the reconstructed frame in Figure 3.11(f) that the salient regions such as part of players are of high resolutions than other non-salient regions. It can be also observed that the background scene of the frame has lower visual quality as it belongs to non-salient region, yet the moving object such as dog is represented with high visual quality due to motion saliency.

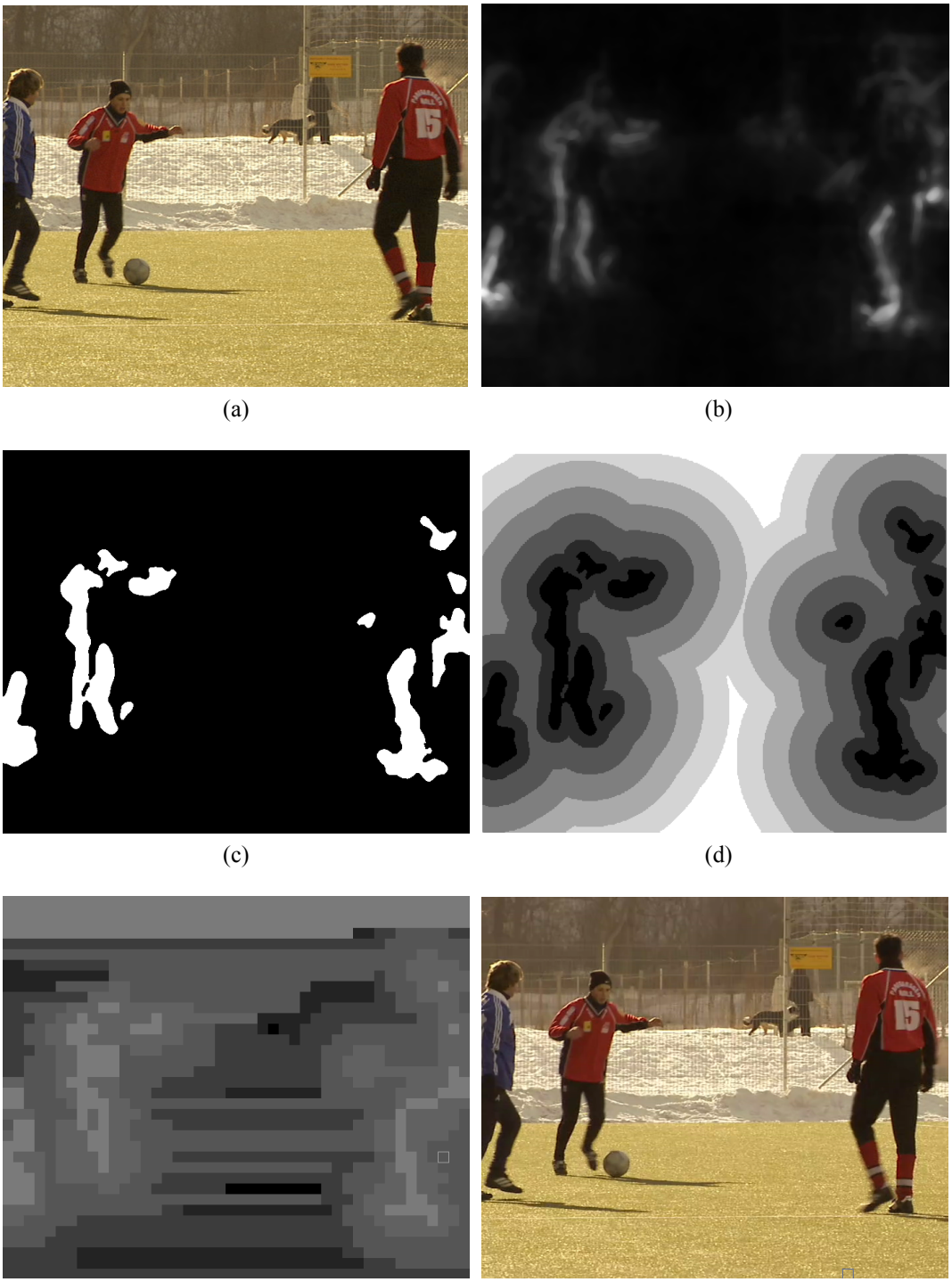
3.5 Experimental Results and Discussion

3.5.1 Experimental results of saliency detection techniques

Experimental Set-up

Various experiments have been carried out to verify the performance of proposed saliency detection schemes. Microsoft Research Asia, Beijing, China (MSRA) image database that contains 1000 real world natural scenes are chosen as test images for different experiments [174].

All experiments are performed using MATLAB version $8.3.0.532$ $(R2014a)$ on an Intel(R)Core(TM)i5-2400 CPU@3.10GHz. Performance of the proposed algorithms are compared against 7 state-of-the-art techniques (IM[64], CB[67], SR[72], FT[74], GB[69], PFT[73] and Pulse-DCT[75]) of saliency detection. All results are compared against ground truth [175] and proposed methods.



(a) (b) (c) (d) (e) (f)

Figure 3.11: Example of proposed foveated video compression for *Soccer* sequence: (a) original frame, (b) saliency map, (c) object map, (d) normalize D_E map, (e) distribution of QP_{fov} (QP values increases from gray to black) and (f) reconstructed foveated video frame

For objective evaluation, a saliency map is converted to a binary object map O which is generated by thresholding saliency map. In the object map white pixels correspond to salient object pixels, while black pixels correspond to the background or non-salient regions. We have proposed to use fuzzy c-means clustering (FCM) [176] to obtain adaptive threshold. In

fuzzy clustering methods, FCM algorithm is the most popular method in image segmentation applications, since it has robust characteristics for ambiguity and can retain much more information than hard segmentation methods [176]. The FCM attempts iteratively to find optimum cluster by minimizing the objective function which is the weighted sum of squared error within group. As $SM = \mathbb{R}^{W \times H} : [0, 1]$, c is the number of clusters, u_{ij} is the degree of membership in i^{th} cluster, m is the fuzzifier set to 2, v_i is the prototype of the centre of cluster i and $d^2(SM_j, v_i)$ is a distance function. The objective function is defined as:

$$J = \sum_{j=1}^{W \times H} \sum_{t=1}^{c} (u_{ij})^m d^2(SM_j, v_i) \tag{3.17}$$

Precision, Recall and F-measure, calculated by (1.11), (1.12) and (1.13), respectively, are chosen as performance metrics to evaluate quantitative performance for salient object detection schemes as preferred by [74, 80]. The receiver operating characteristics (ROC) curve is another benchmark metric for performance evaluation of a decision system, recommended by various salient object techniques [37, 69, 70, 73, 77]. It validates the performance of saliency detection schemes for eye-gaze prediction by measuring accuracy of predictions for fixation and non-fixation regions.

Performance of fusion methods

Various experiments have been performed in order to select an optimum fusion method for maximum performance in saliency detection. Based on four fundamental fusion methods, we have explored best eight combinations of fusion methods. Firstly, a fusion method is applied to inter-scale saliency maps S_n to generate interim saliency map for each colour channel ($\bar{S}^L, \bar{S}^a$ and $\bar{S}^b$). At a later stage, another fusion method is employed on $\bar{S}^L, \bar{S}^a$ and $\bar{S}^b$ to determine master saliency map SM. The eight combinations of fusion methods are: (1) average- average, (2) maximum selection- average, (3) average- local maximum variance, (4) average- maximum selection, (5) maximum selection- maximum selection, (6) maximum selection- local maximum variance, (7) local maximum selection- maximum selection and (8) local maximum selection- local maximum variance.

Figure 3.12 shows the master saliency maps detected by these eight combinations of fusion methods for subjective evaluation. It is clearly visible that fusion methods play major role in determining saliency map in multi-scale analysis. It is observed that contrast of saliency map generated by averaging fusion method is less as compared to other saliency maps. Some noise is also visible in background regions. Local maximum variance results dark patches around salient objects. On the other hand, local maximum selection results lighter patches around prominent areas. It is observed that the combinations of fusion methods: average- average and average- maximum selection yield results fairly close to ground truth.

Figure 3.13 presents comparative analysis of eight combinations of fusion methods in

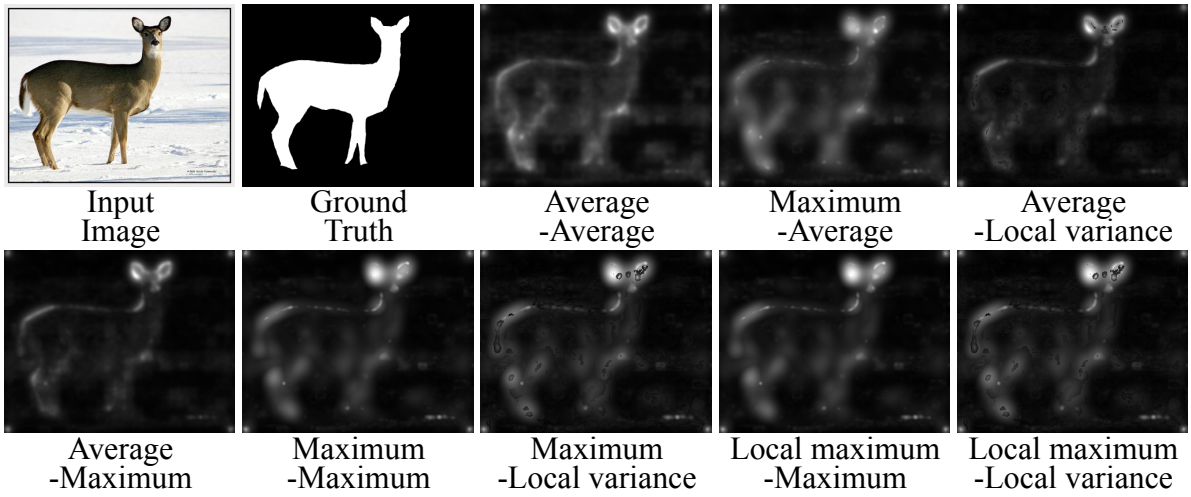


Figure 3.12: Subjective evaluation of saliency maps obtained by applying 8 combinations of fusion methods on inter-scale saliency maps and colour channel saliency maps, respectively

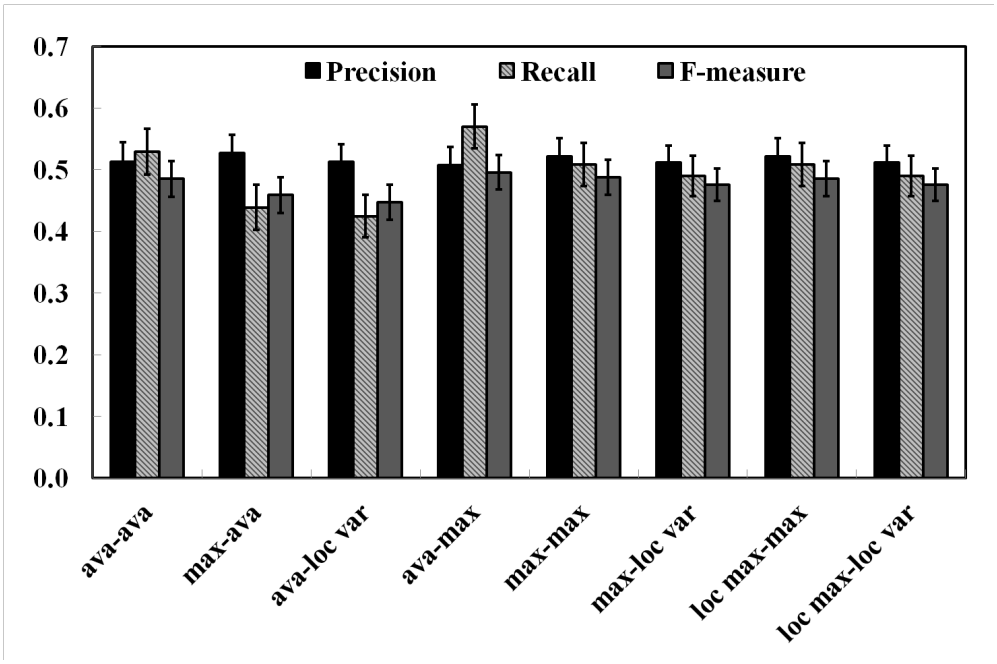
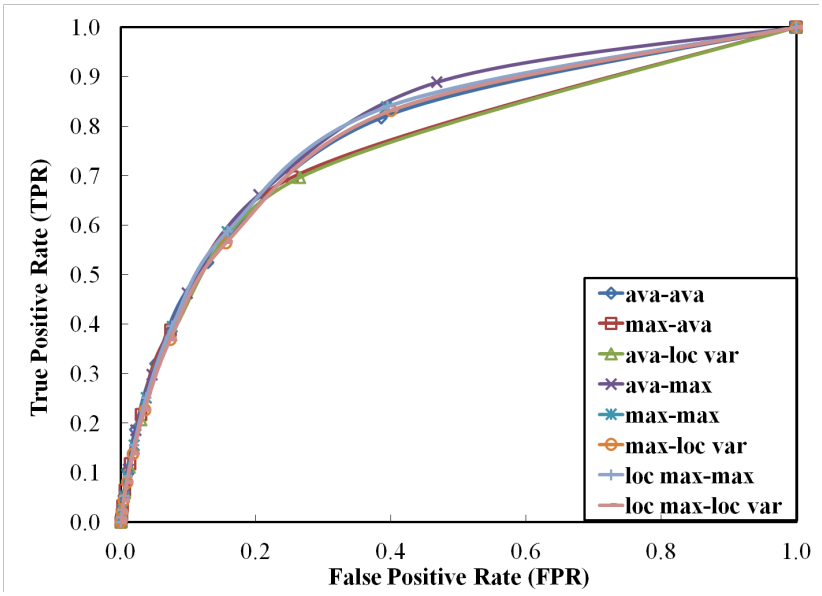


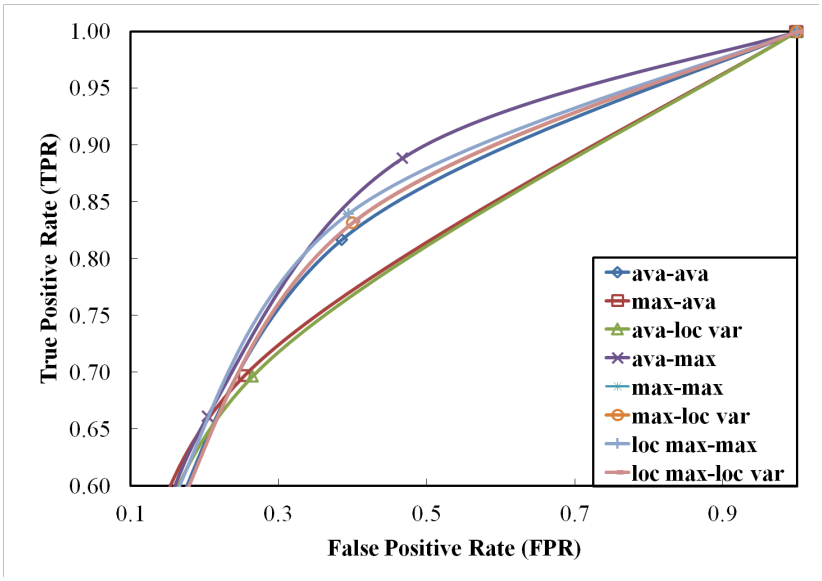
Figure 3.13: Performance comparison of different fusion method combinations based on precision, recall and F-measure for saliency detection with 95% confidence interval

terms of precision, recall and F-measure with 95% confidence interval. It is observed from Figure 3.13 that the precision values of all fusion methods are maintained nearly at the same level but average- maximum selection has higher recall value that leads to higher F-measure value.

In order to assess eye-fixation measure, ROC plots are evaluated for above mentioned fusion methods. Figure 3.14 depicts the comparative performance of ROC-curves and AUC values of these ROC-curves are summarized in Table 3.1. It can be seen that again average-maximum selection fusion method has outperformed other methods with AUC value of 0.7971.



(a)



(b)

Figure 3.14: (a) Comparative performance of receiver operating characteristics (ROC) of fusion methods for saliency map generation, (b) zoomed version of (a)

The subjective evaluation, from Figure 3.12 as well as the objective evaluations, from Figure 3.13 and Figure 3.14, signify the superior performance of average-maximum selection fusion method compared to any other methods. Therefore, we propose that the average-maximum selection combination of fusion method is the best choice for multi-scale analysis of saliency detection. Henceforth, when we compare our proposed FTPBSD scheme, against other state-of-the-art schemes for saliency detection, average- maximum selection combination of fusion method is considered. The mathematical realization of determining saliency map SM using proposed combination of fusion methods would be given as:

Table 3.1: Comparative performance of fusion methods based on AUC metric

Fusion methods (Inter-scale/ colour channels)	AUC
average- average	0.7782
maximum selection- average	0.7512
average- local maximum variance	0.7463
average- maximum selection	0.7971
maximum selection- maximum selection	0.7870
maximum selection- local maximum variance	0.7783
local maximum selection- maximum selection	0.7872
local maximum selection- local maximum variance	0.7784

Table 3.2: Performance comparison of the proposed SDCTPBSD scheme for different averaging methods

Method	Precision	Recall	F-measure	AUC
Arithmetic	0.680	0.526	0.607	0.9032
Geometric	0.726	0.638	0.694	0.9131
Harmonic	0.761	0.702	0.739	0.9192

$$SM = max(\bar{S}^L, \bar{S}^a, \bar{S}^b) \tag{3.18}$$

where $\bar{S}^L$, $\bar{S}^a$, and $\bar{S}^b$ are determined as:

$$\bar{S}^c = \frac{\sum_{n=0}^{N-1} S_n}{N} \tag{3.19}$$

where c represents L^*, a^* and b^* colour channels.

Performance evaluation on static images

In FTPBSD, with the help of performance analysis of fusion methods, inter-scale saliency maps are combined using *average or mean* operation and *max* operation is performed on colour channel saliency maps to generate master saliency map. In SDCTPBSD, we have explored further and have determined an optimum mean method out of arithmetic mean, geometric mean and harmonic mean. Harmonic mean does not get affected much due to fluctuation in sampling values and unlike arithmetic mean, it gives less weight to high valued outliers and yields true average. Therefore, it is proposed to use harmonic mean to combine inter-scale saliency maps in SDCTPBSD. A summarized performance analysis is given in Table 3.2.

Table 3.3 summarizes the performance of saliency detection schemes in terms of precision, recall and F-measure on MSRA image database. Figure 3.15 exhibits performance comparison of the proposed method against other saliency detection methods in terms of precision, recall and F-measure with 95% confidence interval. It is observed that FT[74],

Table 3.3: Average precision, average recall, average F-measure and average area under the curve (AUC) values of proposed schemes and other existing schemes

Schemes	Precision	Recall	F-measure	AUC
FT[74]	0.721	0.554	0.596	0.8281
GB[69]	0.606	0.568	0.538	0.8439
IM[64]	0.679	0.113	0.237	0.6394
CB[67]	0.501	0.427	0.445	0.7595
SR[72]	0.473	0.222	0.310	0.6580
PFT[73]	0.489	0.490	0.453	0.7792
Pulse-DCT[75]	0.515	0.517	0.479	0.7843
FTPBSD	**0.508**	**0.570**	**0.496**	**0.7971**
SDCTPBSD	**0.761**	**0.702**	**0.739**	**0.9192**

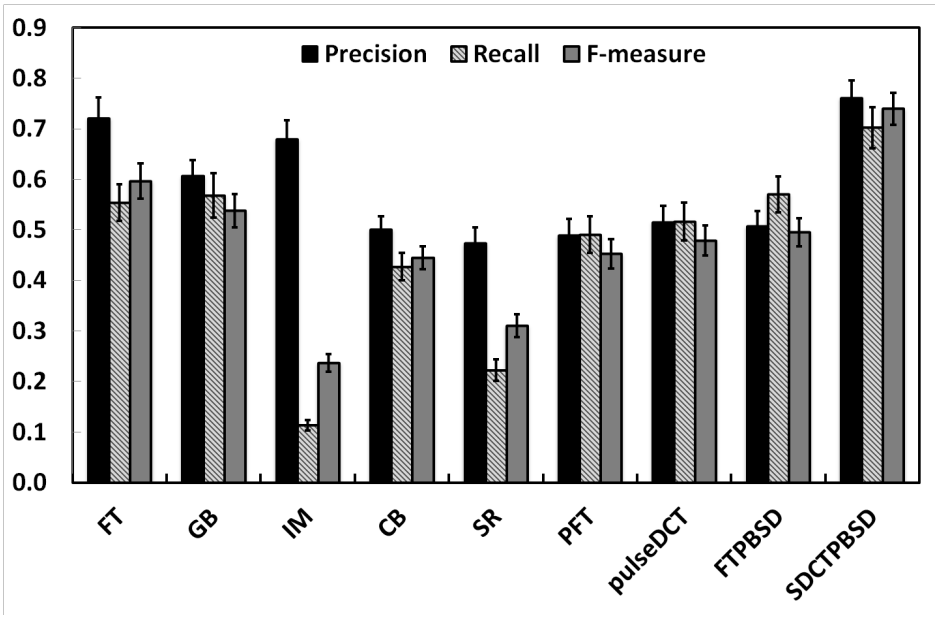
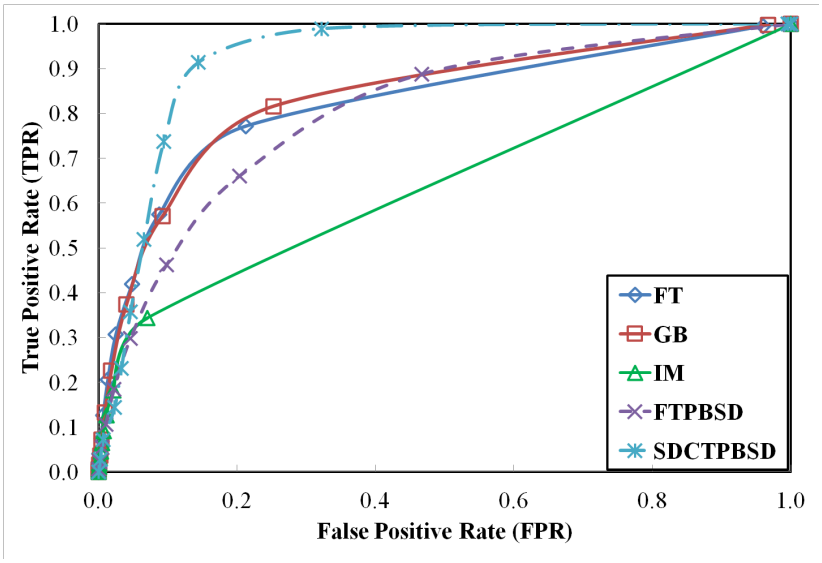


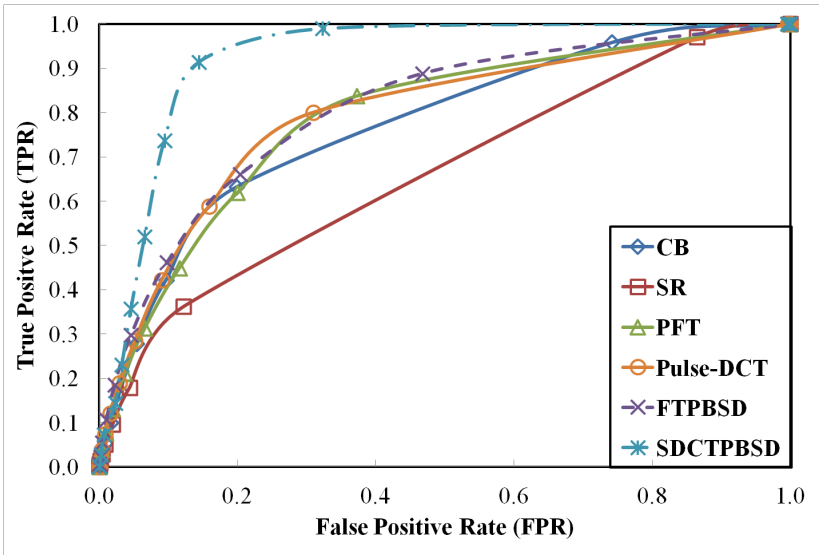
Figure 3.15: Performance comparison of different schemes based on precision, recall and F-measure for saliency detection against the proposed methods with 95% confidence interval

GB[69] and IM[64] have good precision values than the other state-of-the-art methods. It indicates that these methods are better suited for saliency detection, but IM[64] exhibit very poor recall value and hence result in poor F-measure value. Only FT[74] and GB[69] present better performance in terms of precision, recall and F-measure. However, Table 3.3 demonstrates that the proposed schemes outperform other existing salient detection schemes in all aspects.

ROC curves of the proposed schemes (FTPBSD and SDCTPBSD) and other seven state-of-the-art methods are shown in Figure 3.16. It is observed that FT[74], GB[69] are leading with AUC value of 0.8281 and 0.8439, respectively. However, it is pointed out that the proposed method SDCTPBSD has higher AUC values than all others. On the other hand, FTPBSD outperforms IM, CB, SR, PFT and Pulse-DCT. The proposed FTPBSD and SDCTPBSD methods have AUC of 0.7971 and 0.9192, respectively as shown in Table 3.3.



(a)



(b)

Figure 3.16: Comparative graphical analysis of receiver characteristics (ROC) : (a) the proposed methods against FT [74], GB [69] and IM [64] and (b) the proposed method against CB [67], SR [72], PFT [73] and Pulse-DCT [75]

For comparative analysis in terms of subjective evaluation, visual results are shown in Figure 3.17. It is observed that FT [74] detects more non-salient regions such as shown in third and fourth columns. GB [69] yields good performance, but sometimes does not detect the complete salient objects as shown in third column. IM [64] also fails to detect complete salient regions and helps in directing visual attentions only. CB [67] and SR [72] highlight minimum details and suppress salient regions due to their biasing towards edges. PFT [73] and Pulse-DCT [75] suffer from ill-shaped boundaries for salient regions. However, it is observed that the proposed schemes not only highlight salient regions almost uniformly, but also are consistent with attentions. For example, the proposed SDCTPBSD scheme

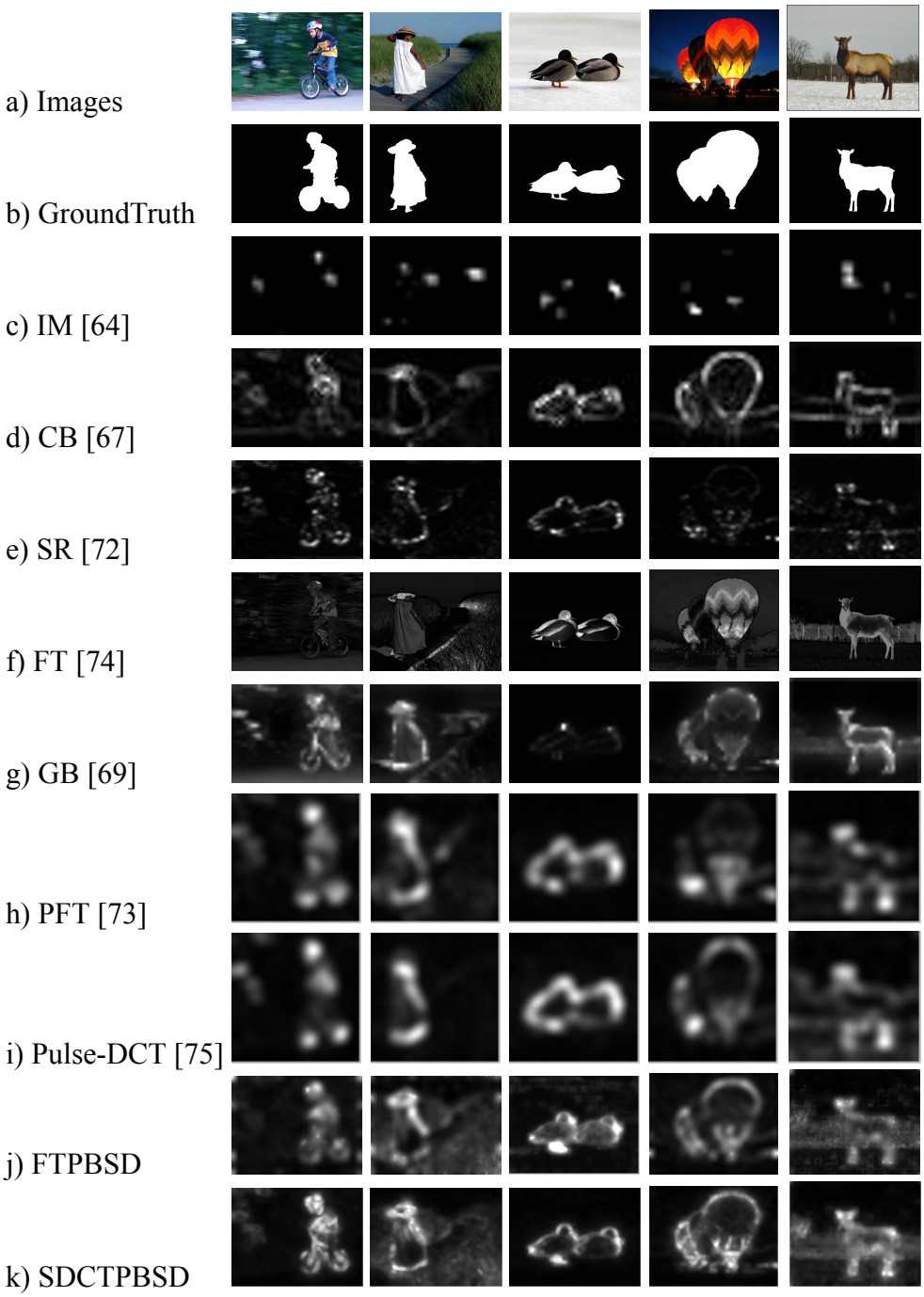


Figure 3.17: Some examples for subjective analysis of saliency detection techniques. The first row contains the test images, second row contains ground truths and all the results of seven state-of-the-arts for saliency detection methods (IM [64], CB [67], SR [72], FT [74], GB [69], PFT [73] and Pulse-DCT [75]) against the proposed methods (FTPBSD and SDCTPBSD) are shown in top to bottom order, respectively

detects the kid and bicycle and have excluded remaining background in first column of Figure 3.17, while others detect kid or bicycle partly and even some background pixels are also considered as salient ones. It is found that the proposed schemes produce more uniform and consistent saliency values inside the objects rather than other existing schemes. However, in case of low contrast images such as shown in fifth column, the performance

Table 3.4: Performance comparison of the proposed SDCTPBSD method against TSR[76] and Pulse-DCT[75] for motion saliency detection on video dataset

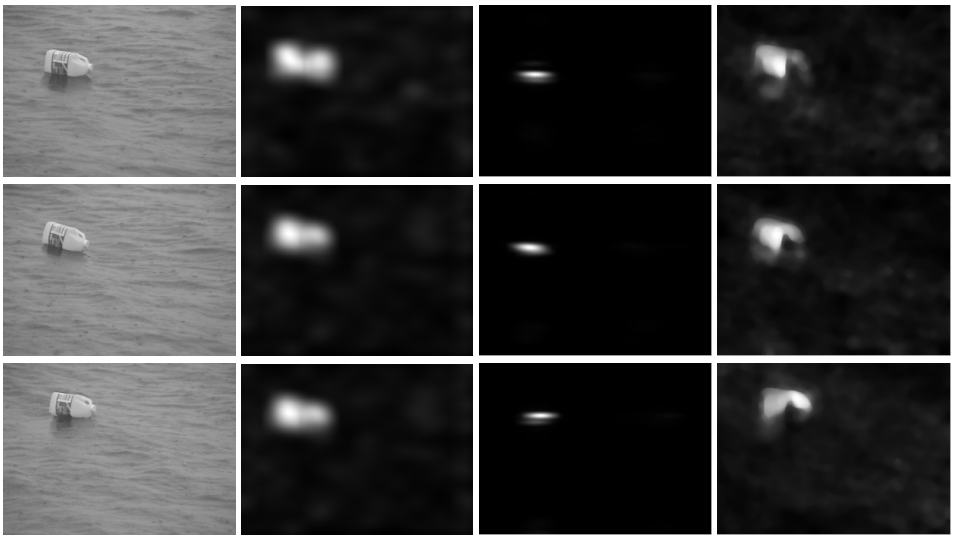
Video Sequences	**TSR[76]**			**Pulse-DCT[75]**			**Proposed Method**		
	Precision	*Recall*	*F-measure*	*Precision*	*Recall*	*F-measure*	*Precision*	*Recall*	*F-measure*
Birds	0.067	0.783	0.096	0.080	0.892	0.109	0.090	0.792	0.125
Boats	0.028	0.868	0.042	0.038	0.976	0.057	0.210	0.705	0.274
Bottle	0.087	0.78	0.124	0.100	0.803	0.142	0.679	0.670	0.676
Cyclists	0.068	0.743	0.097	0.077	0.928	0.112	0.203	0.774	0.269
Freeway	0.008	0.014	0.012	0.008	0.124	0.013	0.306	0.260	0.288
Hockey	0.125	0.709	0.172	0.195	0.788	0.257	0.376	0.405	0.352
Jump	0.198	0.894	0.267	0.228	0.941	0.296	0.375	0.730	0.354
Ocean	0.077	0.536	0.108	0.087	0.616	0.122	0.100	0.913	0.141
Peds	0.191	0.556	0.245	0.221	0.607	0.280	0.389	0.429	0.390
Surfers	0.028	0.830	0.041	0.038	0.996	0.057	0.314	0.846	0.374
Chopper	0.140	0.530	0.186	0.180	0.740	0.220	0.117	0.934	0.163
Flock	0.236	0.676	0.301	0.286	0.763	0.359	0.145	0.644	0.194
Skiing	0.057	0.830	0.082	0.072	0.953	0.104	0.107	0.980	0.151
Surf	0.004	0.909	0.006	0.008	0.887	0.013	0.015	1.000	0.023

of the proposed schemes are not highly promising. The low contrast boundary between salient objects and the surrounding background regions make detection of salient objects very difficult. By and large, from subjective and objective analysis, it is observed that proposed schemes outperform the other existing schemes and they are promising candidates for saliency detection as well as eye-gaze fixation.

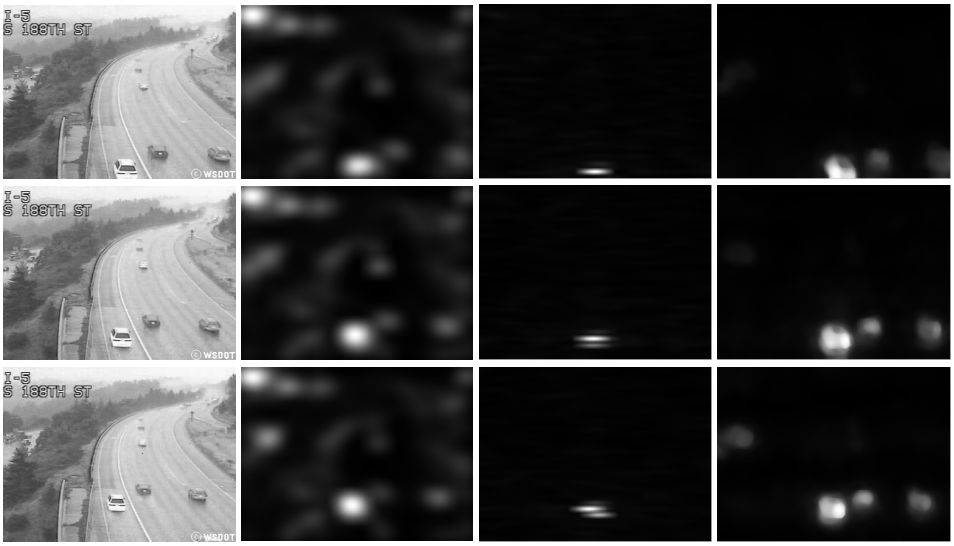
Performance evaluation on video sequences

The performance of the proposed SDCTPBSD scheme is evaluated with a dataset of 14 video sequences obtained from [78]. The performance of the SDCTPBSD is compared against Pulse-DCT [75] and temporal spectral residual (TSR) [76]. Precision, recall and F-measure ($\alpha = 0.5$) are computed and tabulated in Table 3.4 for quantitative analysis. It is observed by *bottle* sequence in Figure 3.18(a) that Pulse-DCT [75] mostly detects the motion saliency points and guides the visual attention but fails to capture whole moving objects and it misses the significant details of moving cars for the *freeway* sequence as shown in Figure 3.18(b). The proposed SDCTPBSD scheme not only detects true motion saliency, but also yields complete contour of salient objects. Since the proposed SDCTPBSD scheme generates finally saliency map from multi-scale architecture and considers all scales information equally important, the scale-invariant property of the proposed scheme produces finer information. On the other hand, Pulse-DCT [75] and TSR [76] resize an input frame to $64-$pixels wide and yields a much coarser information in saliency map.

The complete spatio-temporal saliency generation results are shown in Figure 3.19 for the *News* video sequence of CIF (288×352) pixels resolution. An original frame of *News* video sequence is shown in Figure 3.19(a). Figure 3.19(b) shows computed spatial saliency map by the SDCTPBSD scheme, covering all the salient regions (two persons are reading news in foreground, one lady is dancing inside background screen and logos present in foreground



(a)



(b)

Figure 3.18: Motion saliency in video sequences: (a) *bottle* and (b) *freeway*. Performance of the proposed method results (fourth column) are compared against TSR[76] (second column) and Pulse-DCT [75] (third column) outcomes

as well as inside the blue screen of the tv in the background) based on intensity and colour channels. Figure 3.19(c) depicts extracted motion saliency map, it can be observed that motion of dancing lady inside the background screen is detected. Finally, the spatio-temporal saliency map is shown in Figure 3.19(d).

3.5.2 Experimental results of foveated video compression in H.264/AVC

To analyse the performance of the proposed FVC schemes, various experiments have been conducted on H.264/AVC joint model reference software (version JM18.6) [138]. For experiments, two versions of FVC schemes are considered due to two proposed saliency

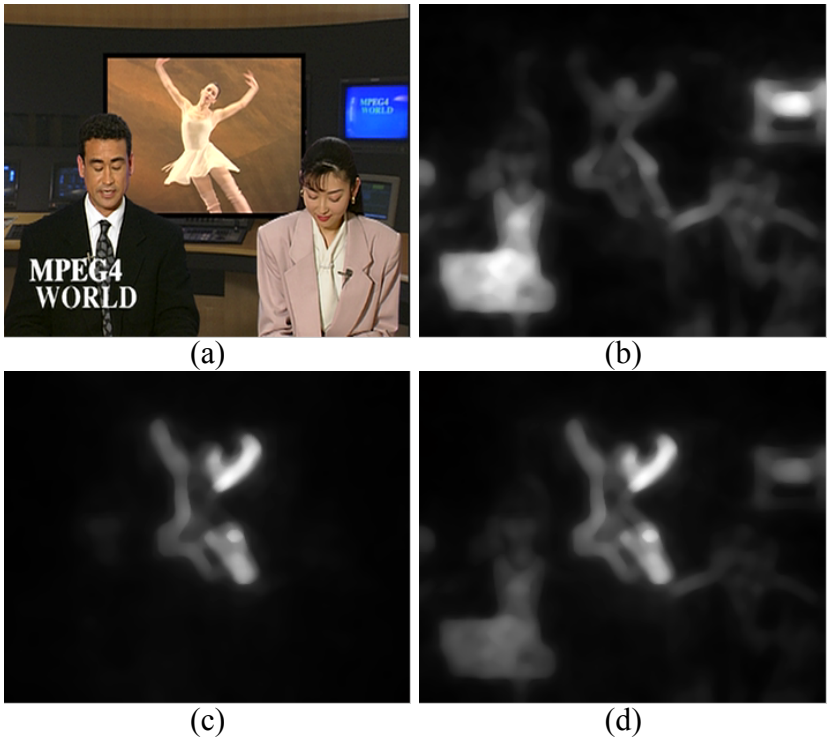


Figure 3.19: Spatio-temporal saliency map in *News* video sequence : (a) input frame, (b) result of spatial saliency map ($S^{spatial}$) of the proposed method, (c) motion saliency map (S^{motion}) generated by the proposed method and (d) finally spatio-temporal saliency map (SM) generated by weighted summation of both spatial and motion saliency maps

Table 3.5: Characteristics of test video sequences

Video Sequence	Foreman	Highway	Mobile	Bus	Crew	Soccer	Old town cross	Park joy
Resolution	144×176 (QCIF)	144×176 (QCIF)	288×352 (CIF)	288×352 (CIF)	576×704 (4CIF)	576×704 (4CIF)	720×1280 (HD-720p)	720×1280 (HD-720p)
Total Frames	300	2000	300	150	600	600	500	500
Frames per second	30	30	30	30	60	60	60	60
Motion Type	Medium	Fast	Medium	Fast	Slow	Fast	Slow	Medium

detection schemes. One is FVC-FTPBSD based on FTPBSD scheme, while another is FVC-SDCTPBSD based on SDCTPBSD scheme. Both of these FVC schemes are compared against conventional uniformly sampled video encoding scheme. All experiments are carried out on standard video sequences like *Foreman, Highway,* etc. Video sequences are categorized in terms of their resolutions as QCIF, CIF, 4CIF and HD 720p. The details of the test video sequences are listed in Table 3.5.

A set of four quantization parameter (QP) values 20, 26, 32 and 38 are used to encode the video sequences. Entropy encoding mode is set to context adaptive variable length coding (CAVLC). To measure visual quality, the cumulative peak signal to noise ratio (CPSNR) is used in our experiments. The detailed encoder configuration for JM 18.6 is listed on Table 3.6.

Table 3.6: Encoder configuration in JM 18.6 reference software of H.264/AVC

Common Parameters	Inter-Coding
FrameRate = 30.0	FramesToBeEncoded = 100
DisableIntra16x16 = 1	IntraPeriod = 0
EnableIPCM = 0	IDRPeriod = 30
NumberBFrames = 0	QPISlice = 26
PicInterlace = 0	QPPSlice = {20, 26, 32, 38}
MbInterlace = 0	DisableSubpelME = 0
RDOptimization = 1	SearchRange = 32
NumberBFrames = 0	ChromaMEEnable = 0
YUVFormat = 1	PSliceSearch4x4 = 1
SourceBitDepthLuma = 8	PSliceSearch8x8 = 1
SourceBitDepthChroma = 8	NumberReferenceFrames = 1
Transform8x8Mode = 1	DisableIntraInInter = 1
SymbolMode = 0	

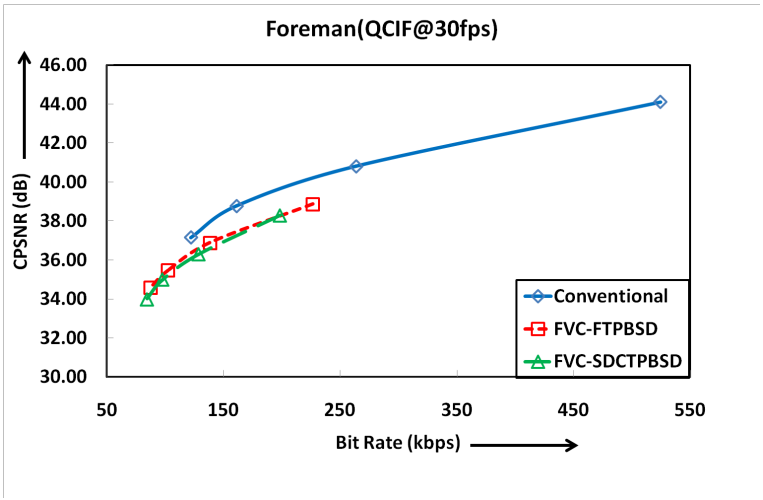


Figure 3.20: Rate-distortion curves for *Foreman* sequence

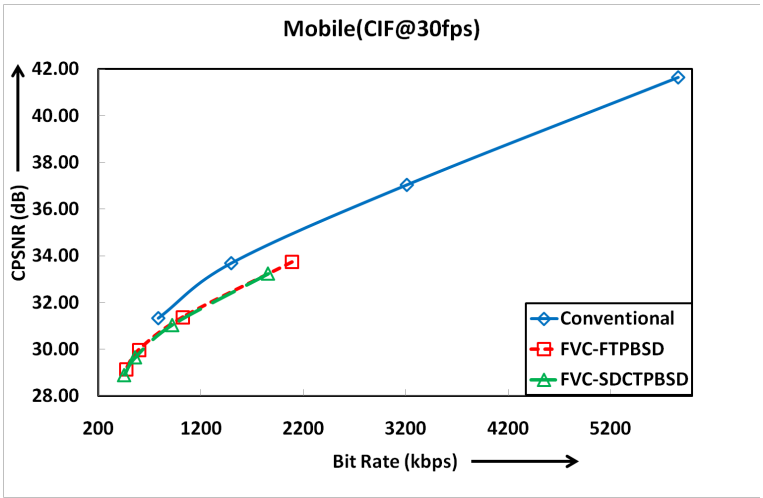


Figure 3.21: Rate-distortion curves for *Mobile* sequence

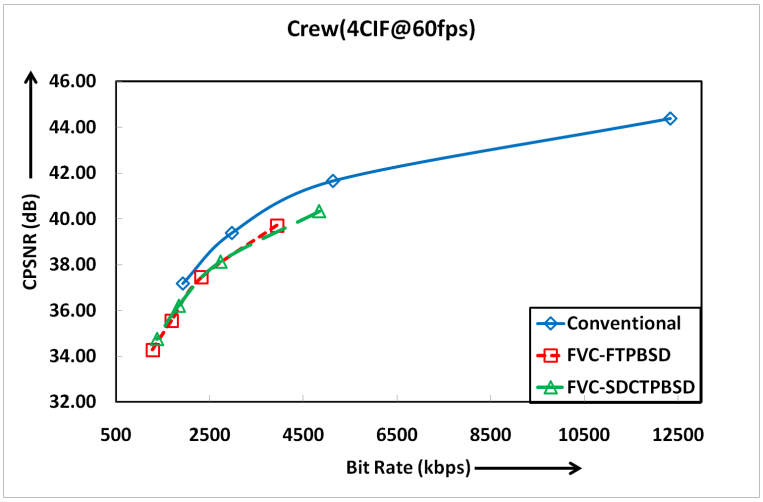


Figure 3.22: Rate-distortion curves for *Crew* sequence

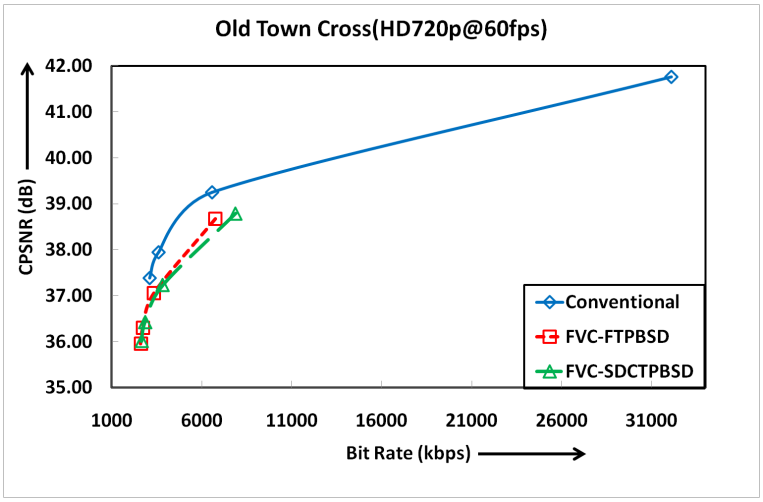


Figure 3.23: Rate-distortion curves for *Old Town Cross* sequence

Experiment 1: Bjontegaard delta bit-rate and Bjontegaard delta PSNR results

In this experiment, both proposed FVC schemes are compared with respect to BD-PSNR and BD-bitrate against the conventional video encoder. The comparative analysis is tabulated in Table 3.7. In Bjontegaard metric positive numbers in BD-PSNR represent gain, while negative numbers in BD-bitrate show reduction in bit-rate. The performance comparisons between conventional video encoder and the proposed FVC schemes in terms of R-D curves of *Foreman, Mobile, Crew* and *Old Town Cross* video sequences are shown in Figure 3.20 through Figure 3.23, respectively.

It may be observed from Table 3.7 that visual quality is compromised to achieve higher compression ratio. The foveated video will have lower visual quality compared to conventional non-fovated uniform sampling video due to non-uniform resolution encoding. However, this non-uniform sampling helps both FVC schemes to exhibit higher compression performance as compared to conventional encoder. The proposed FVC-FTPBSD shows

Table 3.7: Bjontegaard metric[36] performance in H.264/AVC platform

	Sequence	FVC-FTPBSD	FVC-SDCTPBSD
BD-PSNR (dB)	Foreman	-1.21	-1.43
	Highway	-1.12	-1.13
	Mobile	-1.08	-1.09
	Bus	-0.91	-1.20
	Crew	-0.69	-0.89
	Soccer	-1.54	-1.51
	Old town cross	-1.32	-1.18
	Park joy	-0.80	-0.76
	Average	**-1.08**	**-1.15**
BD-bitrate (%)	Foreman	30.01	33.47
	Highway	51.82	54.43
	Mobile	39.40	37.74
	Bus	28.24	41.23
	Crew	15.92	21.69
	Soccer	49.88	52.79
	Old town cross	32.80	50.12
	Park joy	33.19	26.53
	Average	**35.16**	**39.75**

degradation in BD-PSNR of 1.08 dB on average as compared to conventional encoder for same bit-rate (or equivalently 35.16% increment in BD-bitrate on average for same PSNR). While, the proposed FVC-SDCTPBSD demonstrates degradation in BD-PSNR of 1.15 dB on average as compared to conventional encoder for same bit-rate (or equivalently 39.75% increment in BD-bitrate on average for same PSNR). It is also noticed that for *Highway* sequence both of the proposed FVC schemes have shown significant outcome. For *Highway* sequence, the proposed FVC-FTPBSD shows degradation in BD-PSNR of 1.12 dB for same bit-rate (or equivalently 51.82% increment in BD-bitrate for equal PSNR), similarly FVC-SDCTPBSD presents degradation in BD-PSNR of 1.13 dB for same bit-rate (or equivalently 54.43% increment in BD-bitrate for equal PSNR).

Experiment 2: Analysis of encoding time complexity

In FVC scheme, QP_{fov} is used to encode each frame with spatially varying resolution. The QP_{fov} may change for each macroblock depends upon the D_E value, therefore, on average QP_{fov} will have higher value than conventional encoding average QP. Consequently, less number of coefficients will be generated and encoded and that lead to reduced encoding complexity. Hence, the FVC scheme will have less encoding time than its counterpart. In this experiment, in order to compare the encoding time complexity, we have calculated encoding time for each candidate encoder: conventional, FVC-FTPBSD and FVC-SDCTPBSD. The

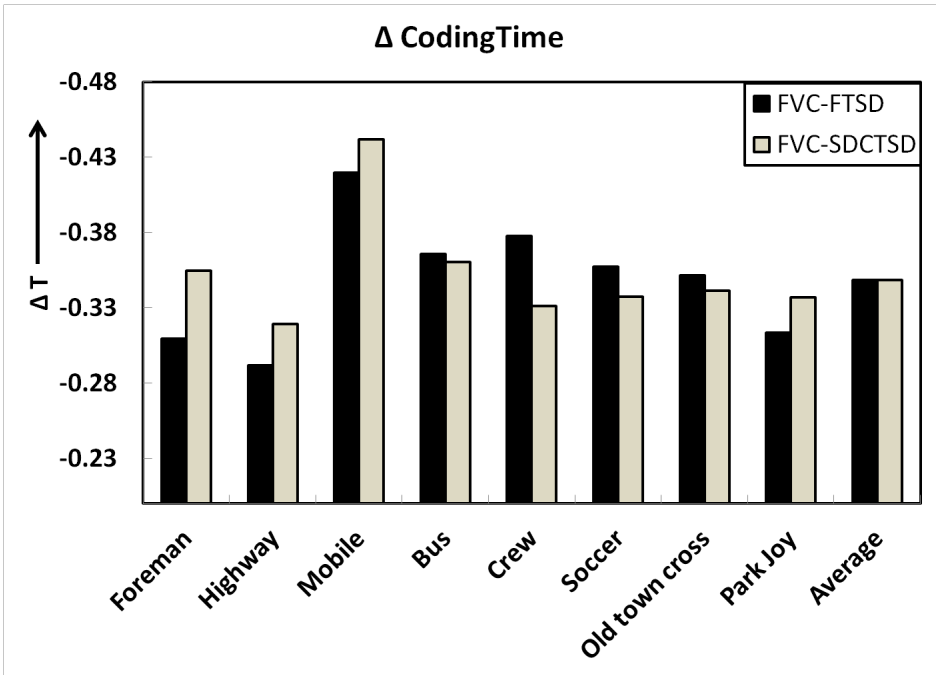


Figure 3.24: Performance comparison of the proposed FVC schemes in terms of Δcoding time with respect to conventional video encoder

relative change in coding time (ΔT) is evaluated as :

$$\Delta T = \frac{T_{proposed} - T_{reference}}{T_{reference}} \tag{3.20}$$

where encoding time for the conventional encoder is considered as reference. The positive numbers represent increase in coding time with respect to conventional encoding and vice-versa. As ΔT represents the relative change in encoding time, so 1.0 represents increment in encoding time by 100%. In other words, it represents 200% encoder run-time ratio. Figure 3.24 demonstrates the relative change in coding time (ΔT) of proposed FVC schemes for various video sequences with respect to conventional video encoder. It is observed that both FVC-FTPBSD and FVC-SDCTPBSD have same average ΔT of -0.35 or encoder time ratio of 65% with respect to conventional video encoder. Therefore, it may be stated that the proposed FVC-FTPBSD and FVC-SDCTPBSD not only achieve higher compression ratio but also take less time to encode the foveated video data.

Experiment 3: Subjective evaluation

To perform the subjective evaluation of the proposed FVC schemes (FVC-FTPBSD and FVC-SDCTPBSD), the results of the proposed schemes are shown against uniformly sampled conventional encoder for *Soccer* sequence in Figure 3.25. The original 63^{th} frame is shown in Figure 3.25(a). Figure 3.25(b) and Figure 3.25(c) show the object maps which are binary saliency maps of proposed FTPBSD and SDCTPBSD saliency detection schemes, respectively. For QP values of 32 and 38, reconstructed frames of conventional,

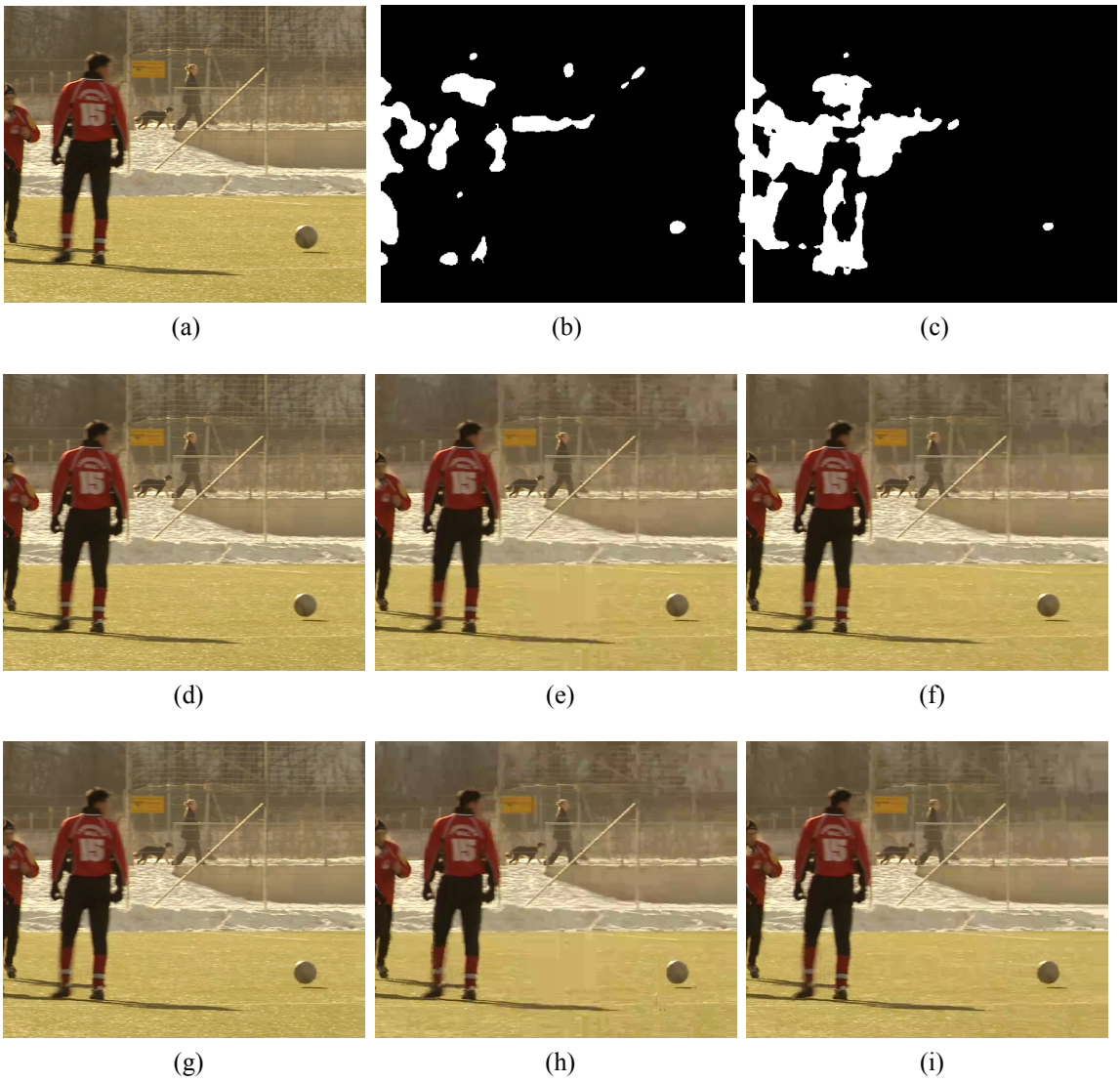


Figure 3.25: Subjective evaluation of proposed FVC schemes for $QP = 32, 38$ for *Soccer* sequence: (a) Original frame, (b) FTPBSD object map, (c) SDCTPBSD object map, (d)-(i) Reconstructed frames:
(d): conventional encoder (209.93 kbps, 40.48 dB),
(e): FVC-FTPBSD encoder (133.77 kbps, 37.69 dB),
(f): FVC-SDCTPBSD encoder (137.89 kbps, 37.87 dB),
(g): conventional encoder (153.5 kbps, 39.08 dB),
(h): FVC-FTPBSD encoder (113.86 kbps, 36.73 dB),
(i): FVC-SDCTPBSD encoder (115.9 kbps, 36.96 dB)

FVC-FTPBSD and FVC-SDCTPBSD encoders are compared. It may be observed that the jacket of the player, soccer and walking woman along-with a dog are salient objects in the scene. The jacket of the player has pop-out because of high colour channel saliency and for other salient objects motion saliency plays a major role. These objects have higher resolutions compared to other objects such as grass in the playground, fencing poles, trees in the background.

It may be observed that reconstructed frames of the proposed schemes (FVC-FTPBSD and FVC-SDCTPBSD) have considerable lower bit-rates with a marginal quality degradation, in terms of PSNR, but with almost similiar visual quality for salient regions and a slight loss in quality for non-salient regions in comparison with a conventional encoder.

In other words, our HVS will appreciate the results yielded by the proposed schemes since we have higher perception for the salient regions compared to non-salient regions. Hence, this figure clearly demonstrates that the proposed schemes yield good visual quaity with better bit-rates (lower bpp). Thus, higher compression performance is achieved with good visual quality.

3.6 Conclusion

In this chapter, we have discussed the non-uniform space-variant resolution property of HVS. Various features, which control the movement of eyes for foveation, are also studied. We have proposed two saliency detection techniques (FTPBSD and SDCTPBSD) to determine the most important object in a scene. Both of these schemes calculate saliency maps in frequency domain with multi-scale analysis. The proposed saliency detection techniques outperform other existing algorithms. The SDCTPBSD yields much higher precision, recall and AUC than FTPBSD. Based on these two saliency detection techniques we have also proposed two FVC schemes known as FVC-FTPBSD and FVC-SDCTPBSD. The proposed FVC schemes greatly reduce the bit-rate of a video data while retaining high visual quality to its salient regions.

Chapter 4

Development of Efficient Directional Transform Schemes

Preview

The 2D-discrete cosine transform (2D-DCT) is widely used in various video coding standards for block based transformation of spatial data. However, for directional featured blocks, 2D-DCT offers sub-optimal performance and may not able to efficiently represent video data with fewer coefficients. To improve the compression ratio further for such directional featured video data, this chapter presents a directional transform scheme based on direction-adaptive fixed length discrete cosine transform (DAFL-DCT) for intra-, and inter-frame. The proposed scheme selects the best suitable transform mode out of eight proposed directional transform modes for each block. In intra-frame coding, 2D-DAFL-DCTs are used whereas conventional 2D-DCT and 1D-DAFL-DCTs are adaptively chosen in inter-frame encoding for each block. In addition, a new modified zigzag scanning pattern is proposed, for 1D-DAFL-DCTs in inter-frame coding, to rearrange these transformed coefficients into a 1D-array, suitable for entropy encoding. The proposed scheme is analysed on JM 18.6 reference software of H.264/AVC platform. The experimental results show that the proposed scheme achieves significant improvement over conventional 2D-DCT and other existing directional transform schemes.

The following topics are covered in this chapter.

- Introduction
- Fundamentals of directional transform
- Direction-adaptive fixed length discrete cosine transform (DAFL-DCT)
- Implementation of DAFL-DCT in H.264/AVC platform
- Experimental results and discussion
- Conclusion

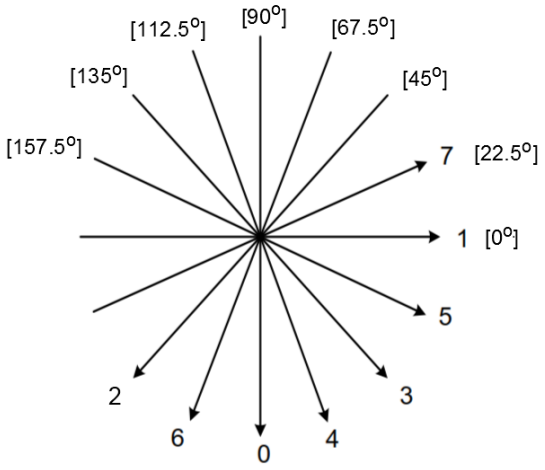


Figure 4.1: Directional angles and corresponding transform modes of DAFL-DCT

4.1 Introduction

DCT based block transform coding is the most popular approach used in image and video coding [6, 13, 15, 21, 95]. It exploits inherent spatial correlation among neighbouring pixels. A superior compression performance is accomplished by efficiently encoding uncorrelated coefficients of highly correlated video data. However, for directional featured block, DCT yields sub-optimal performance and generates a large number of non-zero coefficients that deteriorates compression ratio [98]. Various directional transform schemes are proposed in literature for efficiently encoding such directional blocks [97–100, 103, 104]. However, it is observed that these directional transform schemes suffer from many issues like 'mean weighting defect', use of a large number of DCTs, use of a number of scanning patterns and so on. In this chapter, we address these issues by presenting an novel direction-adaptive fixed length discrete cosine transform (DAFL-DCT) scheme to enhance compression performance of directional block. Two new sets of eight DAFL-DCT directional transform modes are proposed; each for 4×4 and 8×8 block. The orientation angles and corresponding directional transform modes (in brackets) are shown in Figure 4.1.

4.2 Fundamentals of Directional Transform

This section presents a brief discussion on sub-optimal performance of conventional DCT for directional blocks followed by a detailed analysis over shortcomings of other existing directional transform schemes available in literature.

4.2.1 Transform coding with correlation model

For a transform coding, a video frame or image is partitioned to non-overlapping finite stationary blocks. Suppose $s_{i,j}$ represents a pixel of i_{th} row and j_{th} column and assumed

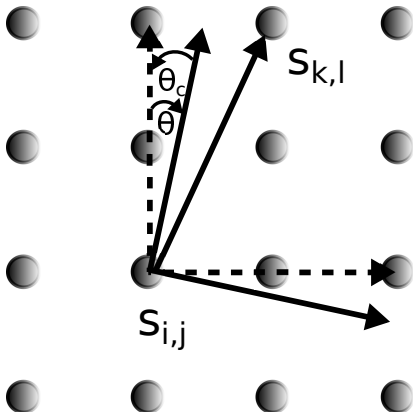


Figure 4.2: Directional image generalized correlation based model for pixels $s_{i,j}$ and $s_{k,l}$, pixels are highly correlated at directional angle θ

to have zero mean and unit variance. Then, the first-order Markov process model [177, 178] is characterized by:

$$s_{i,j} = \rho_1 s_{i-1,j} + \rho_2 s_{i,j-1} - \rho_1 \rho_2 s_{i-1,j-1}, \quad 0 < \rho_1, \rho_2 < 1 \tag{4.1}$$

where ρ_1 and ρ_2 are vertical and horizontal directional correlation coefficients of neighbouring pixels. The inter-pixel correlation matrix of the model is defined by:

$$E_v[s_{i,j} s_{k,l}] = \rho_1^{|i-k|} \rho_2^{|j-l|}, \quad i, j, k, l \in \{0, 1, \cdots, N-1\} \tag{4.2}$$

The Karhunen-Loeve transform (KLT) uses covariance matrix of the given block and thus yields uncorrelated coefficients [178]. The KLT offers optimum performance by packaging most of the energy of the given block to a few coefficients. However, a separate KLT matrix, for each block, limits its practical implementation in various applications [178, 179]. Ahmed et al. [179] have shown that DCT is a close alternative of KLT for a block with high correlation coefficients and its basis functions are independent of the given data. An N-point DCT is defined [179]:

$$F(u) = C(u) \sum_{i=0}^{N-1} f(i) \cos\left[\frac{u\pi}{N}(i + 0.5)\right], \; u \in \{0, \ldots, N-1\} \tag{4.3}$$

where C(u) is a weighting factor given by:

$$C(u) = \begin{cases} \sqrt{\frac{1}{N}}, & u = 0 \text{ or } N \\ \sqrt{\frac{2}{N}}, & \text{otherwise.} \end{cases} \tag{4.4}$$

Similarly, a 2D-DCT is, an extension of 1D-DCT in two-dimensions, defined by:

Figure 4.3: Directional orientation of 8×8 blocks for *Foreman* video sequence: a) original frame, b) red lines represent orientation of 8×8 blocks except $0\,\mathrm{deg}$ and $90\,\mathrm{deg}$, c) applied DAFL-DCT transform modes and d) zoomed version of c)

$$F(u,v) = C(u)C(v)\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} f(i,j)\cos\left[\frac{u\pi}{N}(i+0.5)\right]\cos\left[\frac{v\pi}{N}(j+0.5)\right],\ u,v \in \{0,\ldots,N-1\} \tag{4.5}$$

Introducing a weighting factor C(u) converts DCT into unitary matrix and therefore, the same DCT kernel is used both in forward and inverse transform. In addition, as 2D-DCT is a separable transform, it can be computed with the row-column decomposition by consequently applying 1D-DCTs row-wise and then column-wise. The order of these two operations does not have influence on final outcome. This makes 2D-DCT as an optimum block transform for horizontal and/or vertical dominant edges or orientated blocks.

4.2.2 Directional features and sub- optimal performance of conventional DCT

Let us consider a 4×4 block illustrated in Figure 4.2. Let us also assume the pixels to be highly correlated along the direction of an angle, θ from the vertical axis. By using generalized correlation based model [178], the inter-pixel correlation matrix between pixels $s_{i,j}$ and $s_{k,l}$ is given by :

$$E_v[s_{i,j}s_{k,l}]_\theta = \rho_1^{\left|\,|i-k|\cos\theta+|j-l|\sin\theta\,\right|}\rho_2^{\left|\,-|i-k|\sin\theta+|j-l|\cos\theta\,\right|},\quad i,j,k,l \in \{0,1,\cdots,N-1\} \tag{4.6}$$

It is observed from (4.6) that correlation matrix is restricted by a directional angle, θ leading to degraded transformation performance. In practice, video frames may contain blocks with directional features. Further, 2D-DCT, due to its natural characteristic, does not perform well for such blocks and produces more number of transformed AC coefficients than it would have produced for vertical or horizontal oriented blocks. Figure 4.3 depicts dominant block orientations of first frame of *Foreman* video sequence. It is noticed that a lot of blocks belong to directional textured regions. So, to achieve higher compression ratio these directional blocks must be efficiently encoded.

H.264/AVC supports various intra-predicted (IP) directional modes (excluding DC

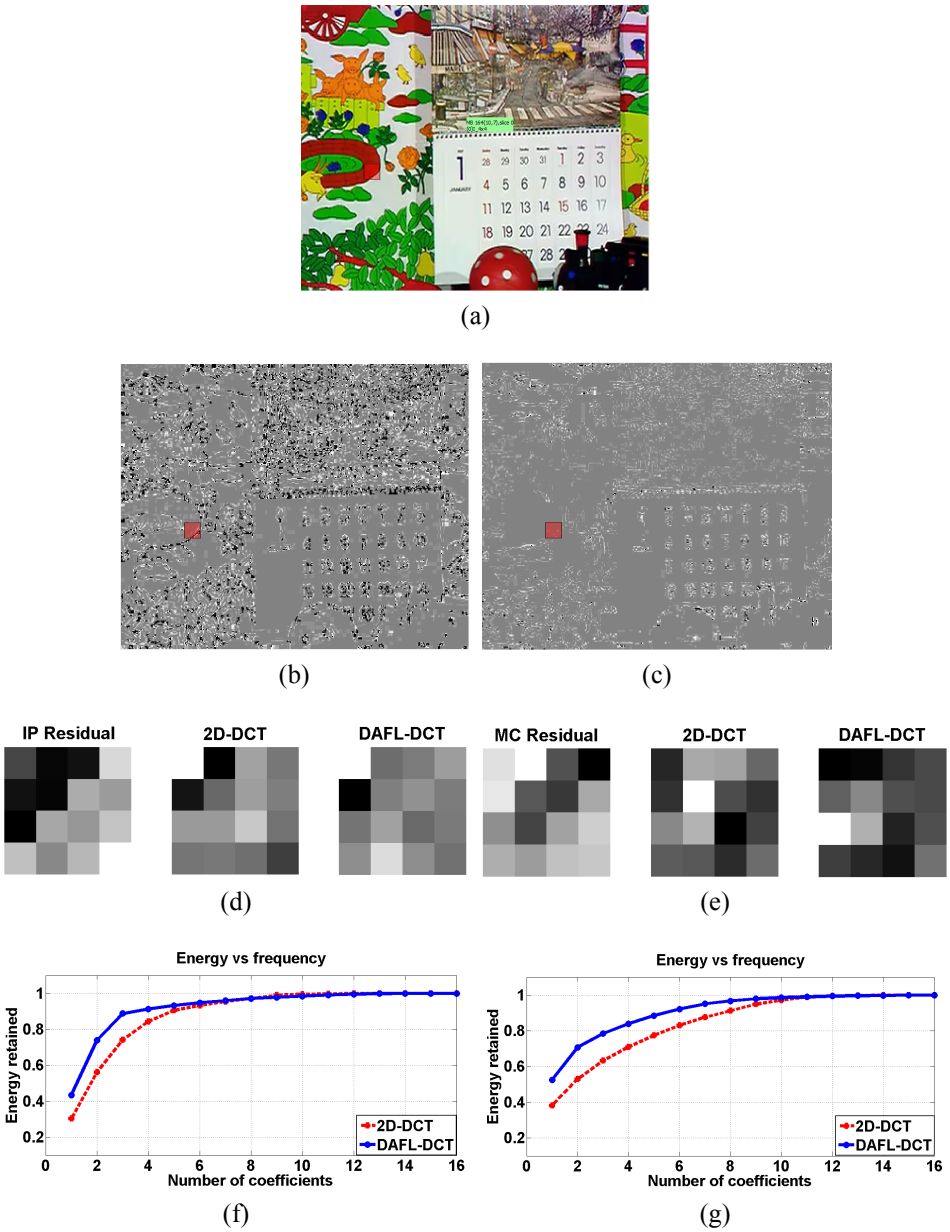


Figure 4.4: Comparison between conventional 2D-DCT and proposed DAFL-DCT for energy compaction: (a) Original frame 002, (b) IP-residual frame 002, (c) MC-residual frame 002, (d) IP-residual block and its 2D-DCT and DAFL-DCT (transform mode 2) transform coefficients, (e) MC-residual block and its 2D-DCT and DAFL-DCT (transform mode 2) transform coefficients, (f) and (g) depict retained energy of transform coefficients for the blocks (d) and (e), respectively

mode) to reduce directional correlation among neighbouring pixels of directional blocks [6]. But, it is found that directionality still exists in IP-residual blocks and that lead to sub-optimal compression performance. A rectangle marked block of a IP-residual frame is shown in Figure 4.4(b). Similarly, another marked block of a a motion-compensated (MC)-residual frame is shown in Figure 4.4(c) for *Mobile* sequence. It is clearly observed from Figure 4.4(d) and Figure 4.4(e) that the blocks still exhibit directional dominance even after being processed either by IP or MC. The resultant transformed coefficients, of both residual blocks

after employing 2D-DCT and proposed DAFL-DCT, are compared for energy compaction in Figure 4.4(f) and Figure 4.4(g). It is found that energy is more concentrated in transformed coefficients of the proposed DAFL-DCT. It is observed that almost 90% of energy is concentrated in the first four or five coefficients of proposed DAFL-DCT, whereas energy is dispersed widely between DC and AC coefficients of conventional 2D-DCT. Hence, these residual blocks can be effectively represented by less number of coefficients using the proposed DAFL-DCT rather than conventional 2D-DCT.

4.2.3 Deficiency of other directional transforms

Recently, many new directional transforms have been developed for video coding that take advantage of directional correlation among pixels to achieve higher coding gain [97–100, 110, 180]. These directional transforms have demonstrated fair coding gain compared to conventional 2D-DCT. Nevertheless, their use in real-time applications are limited and are hard to adopt by various existing video coding standards. The key issues are:

- Mean weighting defect —Most of the transforms [98–100] rearrange block along various directional lines and then perform 1D-DCT. But, the resultant DC coefficients of all directional lines are differently weighted, and hence, they would produce unnecessary non-zero AC coefficients when secondary DCT would be performed. It is known as 'mean weighting defect' and is caused by different weighting factors $C(u)$ [refer (4.4)] as each directional line has different length N-point DCTs. To tackle this problem, DC separation and ΔDC correction modules are required [181], which mean an extra overhead to implementation complexity.

- Multiple length DCTs:

  - Usage of a large number of multiple length directional 1D-DCTs are common. Sometimes, the lengths are even more than the block size N and sometimes too short, for instance, just 1-or 2 [98, 100].

  - Moreover, many directional DCTs do not have transform lengths of integer powers of 2, which limit implementation of a number of fast algorithms for N-point DCT.

  - Extensive use of variable length directional DCTs not only increases computation time, but also produces a large number of DC coefficients that degrades compression ratio. For instance, DDCT uses 15 primary DCTs for 8×8 block in diagonal down-left transform mode [98].

- Computational complexity —Introducing a number of new directional scanning patterns that coincide with the characteristics of directional transform mode for efficient entropy encoding [99, 100] leads to higher implementation cost with more

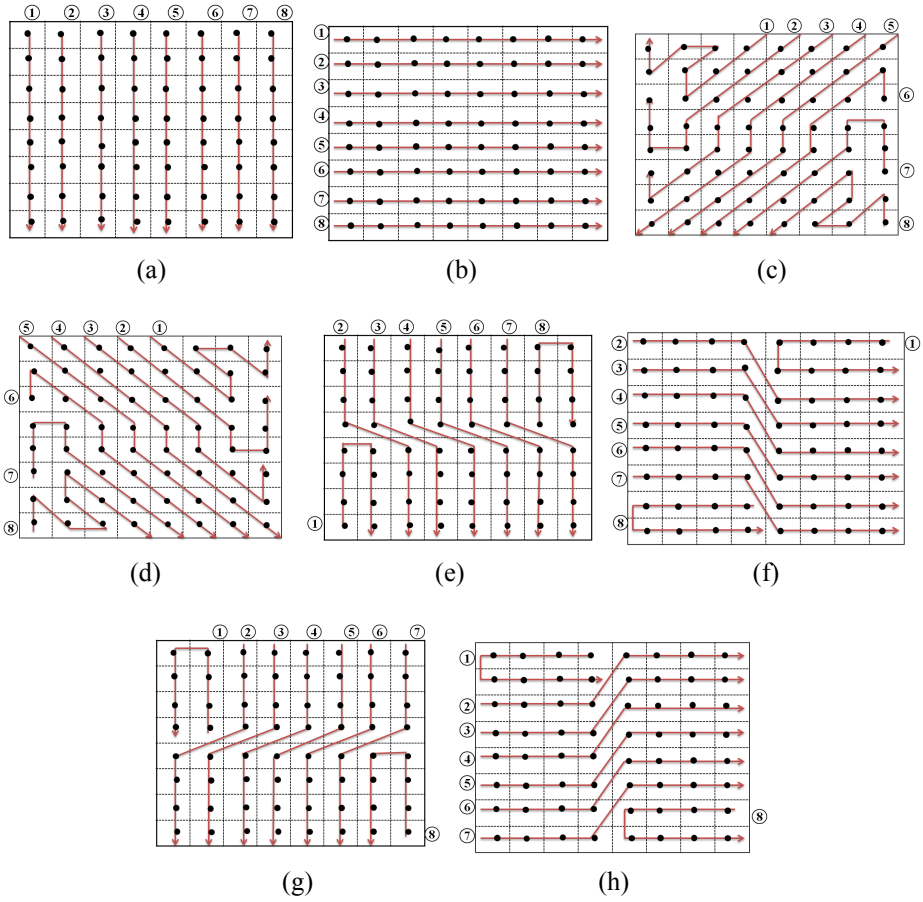


Figure 4.5: DAFL-DCTs for 8×8 blocks

storage capacity and higher computational complexity for selecting these scanning patterns and eventually increases the encoding time.

4.3 Development of Direction-Adaptive Fixed Length Discrete Cosine Transform (DAFL-DCT)

For a directional block, correlation matrix given by (4.6), is restricted by a directional angle θ. To maximize the performance of the correlation matrix (4.6), we counter-rotate a given block by a angle θ_c and hence, the effective rotation angle becomes $\theta - \theta_c$. If θ equals to θ_c, the correlation matrix will be in the form of (4.2) and the resultant transform will yield near optimal performance of KLT. In this section, we develop a new direction-adaptive fixed length transform (DAFL-DCT) to achieve higher compression ratio in case of directional featured blocks. There are two sets of eight directional transform modes, specifically designed for 8×8 blocks and 4×4 blocks. The proposed 8×8 DAFL-DCTs and 4×4 DAFL-DCTs are shown in Figure 4.5 and Figure 4.6, respectively. In each DAFL-DCT transform mode, the directional lines are almost unidirectional and represent dominant orientation of a block. The orientation angles and corresponding transform modes ranging

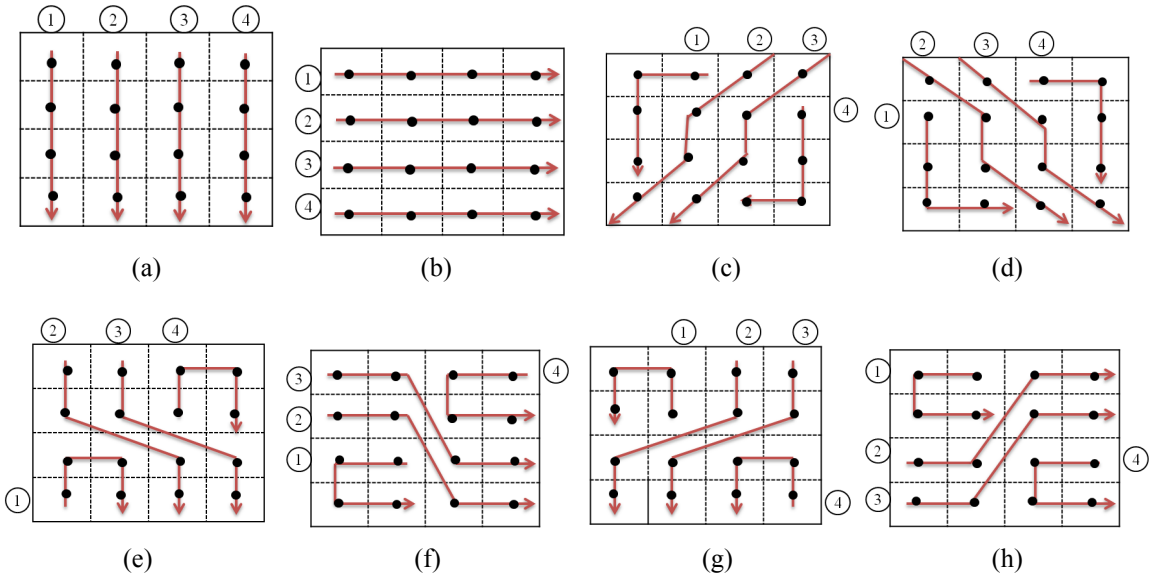


Figure 4.6: DAFL-DCTs for 4×4 blocks

from 0 through 7 for DAFL-DCT are shown in Figure 4.1.

It is observed that the proposed work is closely related to other existing directional transforms that include Directional Discrete Cosine Transform (DDCT) proposed by Zeng et al. [98], Direction-Adaptive Residual Transform (DART) proposed by Cohen et al. [99], and 1D-Transform proposed by Kamisli et al. [97]. Both DDCT and DART have eight directional transform modes similar to 2D-DAFL-DCT and are used in both intra-, and inter-coding. Unlike other methods, 1D-Transform consists of sixteen 1D-DCT directional transform modes and additionally includes conventional 2D-DCT. The proposed work uses only eight 1D-DAFL-DCT transform modes and conventional 2D-DCT for non-directional blocks in inter-coding. There exists three major differences between the proposed DAFL-DCT and other existing directional transforms.

The very first difference lies in the length of the transforms used. If the given block size is $N \times N$, then the proposed DAFL-DCT offers transforms of fixed length N, whereas DDCT, DART and 1-D Transform have multiple transforms of variable length ranging from 1 to $2N-1$, $N/2$ to $2N-1$ and 2 to N, respectively. Secondly, the proposed DAFL-DCT rearranges the pixels according to the selected transform modes and then applies 2D-DAFL-DCT for intra-coding or 1D-DAFL-DCT for inter-coding. The other directional transforms DDCT, DART and 1D-Transform apply directional DCTs to block data directly. The third difference lies in choosing the directional paths of DAFL-DCT transform modes. The directional paths are heuristically designed for fixed length transforms and those paths are different from other existing directional transforms except DAFL-DCT transform modes 0 and 1.

Instead of discussing some specific block sizes like 8×8 or 4×4, let us consider a general block $f_b(r, c)$ of size $N \times N$. The detailed implementation process is discussed as

follows.

Step 1 Directional transpose:

First, out of all available directional DAFL-DCT transform modes (ϖ), where $\varpi = 0, 1, \cdots, 7$ for intra-frame coding and for inter-frame coding additionally include 2D-DCT mode, an optimum directional DAFL-DCT mode is selected for the current block. The selection process of transform modes will be discussed in detail later. After the DAFL-DCT mode is selected, the $N \times N$ block data are rearranged into N 1D-vectors by traversing N directional paths (shown with red lines in Figure 4.5 and Figure 4.6) and arrange them row-wise according to circled row numbers. The length of each directional pattern is N-points. The directional transposed block is a group of row vectors P and expressed as:

$$\bar{f}_b(r, c) = [P_k]', \quad k = 0, 1, \cdots, N-1 \tag{4.7}$$

Step 2 Horizontal 1D-DCT:

The directional transpose has converted a directional block into a horizontally orientated one. Now, N-point 1D-DCT is performed to the reordered block $\bar{f}_b(r, c)$ along the horizontal direction to all rows individually. The horizontal 1D-DCT exploits spatial correlation among directionally oriented block data. The horizontally transformed coefficients are represented as:

$$F_b(u) = [P(u)]', \quad u = 0, 1, \cdots, N-1 \tag{4.8}$$

Step 3 Vertical 1D-DCT:

The horizontal 1D-DCTs have placed the DC coefficients to first column of each row and then AC coefficients in subsequent columns. Now, secondary N-point 1D-DCT along the vertical direction is performed to each column. This process yields one DC coefficient in top-left corner and AC coefficients in remaining $N^2 - 1$ indices. The transformed coefficients are expressed as:

$$F_b(u, v) = [p_{u,v}]_{N \times N}, \quad u, v = 0, 1, \cdots, N-1 \tag{4.9}$$

These coefficients, after quantization, are rearranged from 2D-block into 1D-vector using scanning pattern, so that they can be efficiently encoded by entropy encoder.

Step 4 Directional transform mode selection:

Finally, a rate-distortion optimization (RDO) [182] is employed to select the best suitable DAFL-DCT transform mode. In RDO, each block is transformed and encoded individually by all available transform modes followed by RD cost (also known

as Lagrangian cost function) calculation. The DAFL-DCT transform mode with minimum RD cost is considered to be an optimum transform mode. The RDO is evaluated by optimizing the cost function, J as given below:

$$J(X, \varpi) = \arg\min_{\varpi}\{D_s(X, \varpi) + \lambda \cdot R(X, \varpi)\} \tag{4.10}$$

where X depicts the current block, ϖ represents DAFL-DCT transform mode, J is RD cost, D_s corresponds to distortion, λ is Lagrangian multiplier and R stands for bit-rate.

The fundamental concept in designing of DAFL-DCT transform modes is that the directional paths should not have unequal lengths and efficacy of DCT should be optimally utilised as well. In DAFL-DCT, equal length N point 1D-DCTs are used and the length (N) is also integral multiple of 2. Hence, it is free from aforementioned 'mean weighting defect' and adaptive to fast algorithms on VLSI architecture.

4.3.1 Residual coding

Intra-frame coding

In H.264/AVC intra-frame coding, blocks are initially predicted by available directional IP-modes to reduce directional spatial correlation between pixels [25]. However, it is observed that the directional features are still present in IP-residuals and exercise of conventional 2D-DCT to these residuals lead to sub-optimal compression performance. We propose to use DAFL-DCT with eight transform modes for such IP-direction dominant residual blocks. The best DAFL-DCT transform mode is selected using RDO. The proposed DAFL-DCT yields a few coefficients to represent these directional blocks and thus, improves the compression ratio.

Inter-frame coding

In video coding, the temporal redundancy between frames are reduced by inter-frame coding. In inter-frame, prediction errors or MC-residuals are significantly high along the moving object boundaries or edges as against smooth regions, therefore, they form unidimensional structures aligned to various directions [97]. The conventional 2D-DCT is not a suitable candidate for these directional 1D-structures. We propose to use 1D-DAFL-DCT to transform MC-residuals, unlike proposed 2D-DAFL-DCT for IP-residuals. However, MC-residual frames, in practice may contain several blocks with no dominant directional features, these blocks are encoded well by conventional 2D-DCT. Therefore, in the proposed scheme, inter-frame coding uses eight directional 1D-DAFL-DCT modes and one conventional 2D-DCT.

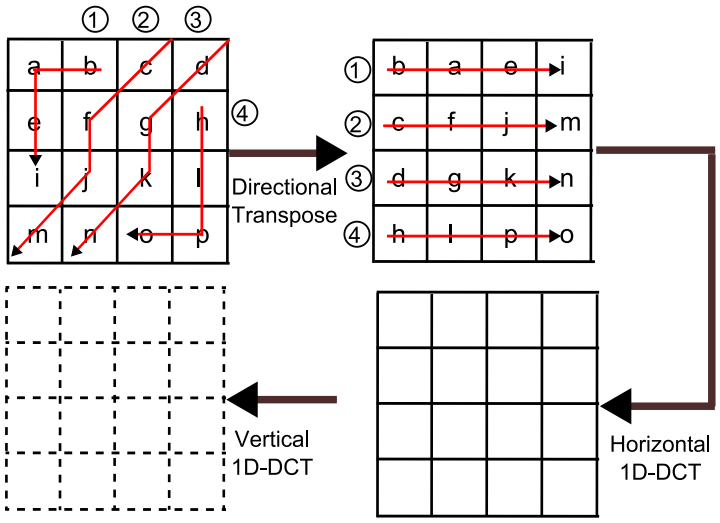


Figure 4.7: Illustration of steps for implementation of DAFL-DCT transform mode 3 for 4×4 block. The dotted block depicts as optional block for MC-residual coding whereas mandatory for intra-frame residual coding

The stepwise implementation of DAFL-DCT for $N \times N$ block is shown in Figure 4.7. Here 4×4 block is considered for illustration. The last stage, shown in dotted line, represents an optional module for inter-frame coding.

An appropriate transform produces less number of coefficients and yields high energy compaction. But, selection of the best suitable DAFL-DCT mode for each block is a challenging task. To resolve this problem, we propose two encoding modes of DAFL-DCT: (a) DAFL-HE and (b) DAFL-LC. The DAFL-DCT encoding modes are discussed in detail below.

4.3.2 DAFL-DCT encoding modes

Direction-adaptive fixed length-high efficiency (DAFL-HE)

The DAFL-HE is a high efficiency mode that selects an optimum DAFL-DCT transform mode for each block. The DAFL-HE is a computational intensive mode, as it uses RDO to choose the best transform mode from all available DAFL-DCT transform modes. In RDO, for every candidate transform mode, the current block is transformed, quantized and entropy encoded to evaluate bit-rate (R) and thereafter, decoding operations are performed to measure distortion (D) of the reconstructed block so that RD-cost can be measured. Finally, a transform mode with minimum RD-cost is selected as an optimum transform mode for the given block.

Direction-adaptive fixed length-low complexity (DAFL-LC)

It is observed that DAFL-HE yields superior compression performance for directional featured blocks, since it selects an optimum DAFL-DCT transform mode at the cost of

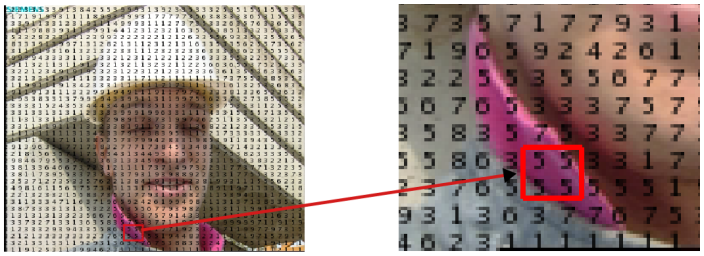


Figure 4.8: Illustration of applied DAFL-DCT transform modes on frame 001 of *Foreman* sequence

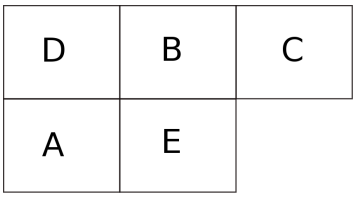


Figure 4.9: Neighbouring blocks

higher complexity. In a low complexity video encoding system, such as a hand-held device that has limited resources, DAFL-HE may become a bottleneck for real-time encoding. Therefore, for resource-constrained systems, we propose a **l**ow **c**omplexity mode called DAFL-LC. DAFL-LC selects the best transform mode from a local set of transform modes made by current block information. The experimental outcomes have shown that DAFL-LC outperforms conventional 2D-DCT in compression ratio with a negligible increase in encoding time and at a marginal compression loss compared to DAFL-HE. In DAFL-LC, the block orientation is assessed by gradient based approach [183]. To select an optimum transform mode, gradient magnitudes and directions of each block data are evaluated and predominant orientation of the block is determined based on orientation histogram. DAFL-DCT transform modes corresponding to dominant directions are taken as candidate transform modes. There are many blocks which do not exhibit governing orientations and selection of appropriate DAFL-DCT modes for such blocks is extremely hard. To mitigate any ambiguity in choosing particular directional transform mode, conventional 2D-DCT is always selected as one of the potential options.

In our experiments, we have also observed that neighbouring blocks tend to opt similar DAFL-DCT transform modes due to high spatial correlation between them. For instance, as shown in Figure 4.8 the neighbouring blocks inside the marked collar area of *Foreman* video sequence have selected the same transform modes. So, unless there is sudden change in orientation or occurrence of edge boundaries, there is high probability that the current block would choose the same transform mode as one of its neighbours. The proposed DAFL-LC exploits spatial correlation among neighbouring blocks to select an optimum transform mode for the current block. The current block E and its neighbours are shown in Figure 4.9. The selection procedure of transform mode for the proposed DAFL-LC is given below.

Step 1: Gradients' magnitude and orientation are evaluated for each sample $s_{i,j}$ of current block E as:

$$Magnitude_{i,j} = |G_{h_{i,j}}| + |G_{v_{i,j}}|, \tag{4.11}$$

$$Orientation_{i,j} = \frac{180^\circ}{\pi} \arctan\left(\frac{G_{v_{i,j}}}{G_{h_{i,j}}}\right) \tag{4.12}$$

where $G_{h_{i,j}}$ and $G_{v_{i,j}}$ represent horizontal and vertical directional gradients, respectively. The gradients are calculate as:

$$G_{h_{i,j}} = s_{i+1,j} - s_{i-1,j}, \tag{4.13}$$

$$G_{v_{i,j}} = s_{i,j+1} - s_{i,j-1} \tag{4.14}$$

Step 2: A histogram is formed for orientation angles covering 0° to 180° by adding block samples weighted by gradient magnitudes. Orientation angles corresponding to highest peak and the next highest are considered as prominent directions of the current block. The DAFL-DCT transform modes corresponding to those directions are chosen as primary candidates (T_{ϖ_1} and T_{ϖ_2}).

Step 3: Conventional 2D-DCT ($T_{\varpi_{2D\text{-}DCT}}$) is a default candidate transform mode in inter-frame coding, whereas the transform mode 0 is employed in intra-frame coding.

Step 4: Transform modes T_{ϖ_A} of adjacent left block (A) and T_{ϖ_B} of top block (B) are also considered as candidate transform modes due to spatial correlation among neighbouring blocks.

Step 5: Finally, RDO is evaluated using (4.10) on a small set of transform candidates determined by local features of current block. The total available transform modes are:

$$\varpi = \{T_{\varpi_1}, T_{\varpi_2}, T_{\varpi_{2D\text{-}DCT}}, T_{\varpi_A}, T_{\varpi_B}\} \tag{4.15}$$

It is clearly observed form the (4.15) that the complexity of the DAFL-LC mode is considerably reduced by generating local data dependent set of limited transform modes. The worst performance of DAFL-LC is observed when an optimum transform mode is chosen such that all the transform modes from set ϖ, as mentioned in (4.15), assumes mutually different values viz. $T_{\varpi_1} \neq T_{\varpi_2} \neq T_{\varpi_{2D\text{-}DCT}} \neq T_{\varpi_A} \neq T_{\varpi_B}$. However, at its worst performance, DAFL-LC requires to evaluate 37.5% less transform modes than DAFL-HE, which reduces the transform evaluation part of encoding time by 37.5%. The experimental results reveal that in real-world video sequences, more than 50% blocks opt same transform modes as their neighbours which effectively reduces the local set of candidate transform

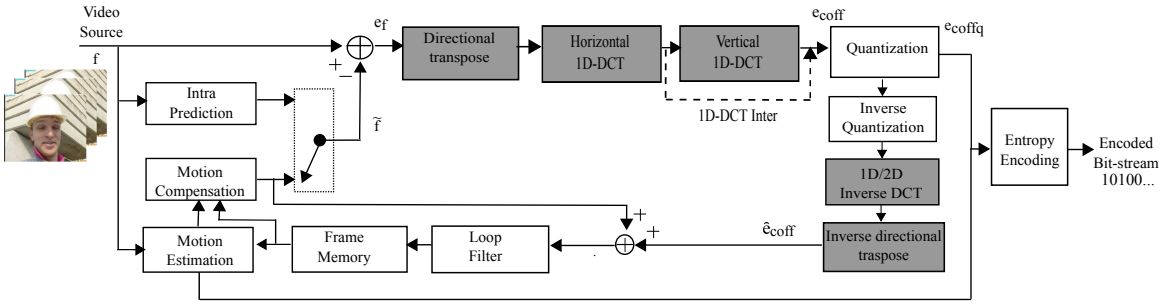


Figure 4.10: Schematic representation of implementation of DAFL-DCT in H.264/AVC video encoder

modes. Hence, unlike DAFL-HE, the number of available transform modes in DAFL-LC would vary from 1 to 5, which significantly reduces DAFL-LC encoding time.

4.4 Implementation of DAFL-DCT in H.264/ AVC platform

H.264/AVC is a leading video coding standard in the current commercial market due to its superior compression performance. To integrate DAFL-DCT in H.264/AVC encoding platform, the conventional two-dimensional integer transform and its inverse is replaced by proposed DAFL-DCT and its inverse, respectively. In addition, there are some other encoding modules that demand meticulous considerations such as entropy coding and coding of side information. The proposed DAFL-DCT integrated into H.264/AVC platform is shown in Figure 4.10.

4.4.1 Entropy coding

H.264/AVC opts either context adaptive variable length coding (CAVLC) or context adaptive binary arithmetic coding (CABAC) to encode transformed and quantized coefficients depending on selected entropy coding mode [6]. The residual blocks are transformed by integer transform, quantized and then entropy encoded. Before applying entropy encoding, the quantized coefficients are first converted from a 2D-block to a 1D-vector using zigzag scanning pattern as shown in Figure 4.11(a). The zigzag scanning pattern exploits the characteristics of 2D-DCT and places higher valued coefficients at the beginning of the vector and keeps lower or zero valued coefficients at the end [27]. Since DAFL-DCT, in intra-frame coding, rearranges residual blocks using directional transpose and then conventional separable 2D-DCT is employed on reordered block, the same conventional zigzag scanning pattern is used.

In inter-frame coding, as discussed in Section 4.3.1, DAFL-DCT uses eight 1D-DAFL-DCT transform modes and one conventional 2D-DCT. The residual blocks that do not exhibit dominant orientations opt conventional 2D-DCT and use the same zigzag

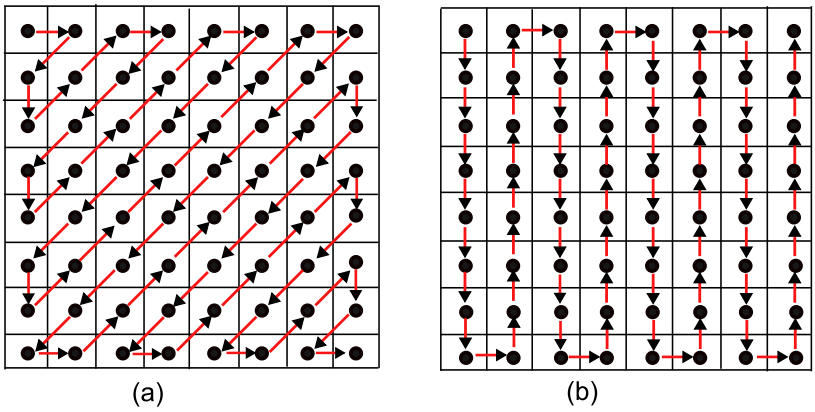


Figure 4.11: Scanning patterns for entropy coding:(a) Conventional zigzag pattern and (b) Modified zigzag pattern

scanning pattern. On the other hand, directional featured residual blocks are transformed by one of the proposed 1D-DAFL-DCT transform modes. The 1D-DAFL-DCT applies horizontal DCT to each row. The DC coefficient of each horizontal DCT are placed in first column of transformed block and remaining AC coefficients are kept in subsequent columns. The conventional zigzag scanning pattern fails to arrange them in expected decreasing order of amplitudes and leads to higher output bit-rate. Therefore, we propose a new modified zigzag scanning pattern, as shown in Figure 4.11(b), for 1D-DAFL-DCT transform modes. The modified scanning order is designed in order to keep high valued DC and low-frequency coefficients at the beginning of the 1D-array and low valued or zero valued high-frequency coefficients at the end.

Let us assume a motion-compensated (MC) residual block of size $N \times N$ in inter-coding. The block data is represented as:

$$X(x,y) = [p_{i,j}]_{N\times N}\,, \qquad i,j = 0,1,\cdots N-1 \tag{4.16}$$

Suppose 1D-DAFL-DCT transform mode 3 is selected for encoding. The given $N \times N$ block pixels are rearranged into N 1D-arrays by traversing N directional lines of the given transform mode as shown in Figure 4.12(a). This vector-set comprises vectors of equal length N arranged row-wise as shown in Figure 4.12(b). The rearranged block is a group of row vectors P, expressed as:

$$\bar{X}(x,y) = [P_k]'\,, \quad k = 0,1,\cdots,N-1. \tag{4.17}$$

To exploit spatial correlation between pixels $p_{i,j}$, N-point 1D-DCT is performed to each row individually as shown in Figure 4.12(c). The horizontally transformed coefficients are represented as:

$$X(u) = [P(u)]'\,, \quad u = 0,1,\cdots,N-1 \tag{4.18}$$

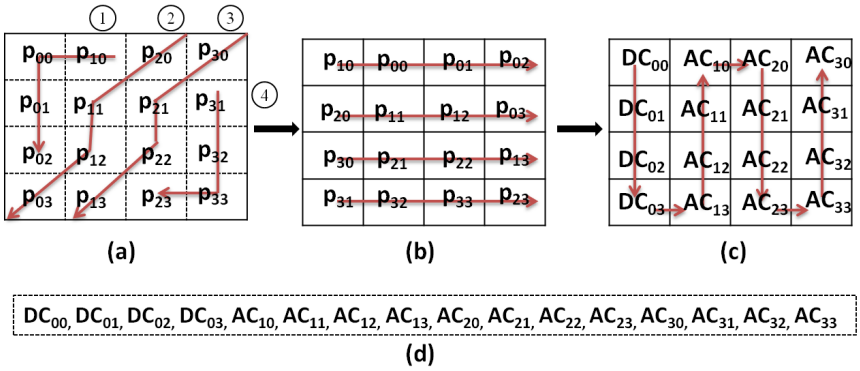


Figure 4.12: Illustration of steps for implementation of DAFL-DCT with modified scanning order for entropy encoding. DAFL-DCT Transform mode 3 is considered for MC-residual block. (a) Original block, (b) Reordered block, c) Transform coefficients after applying horizontal 1D-DCT to each rows, and (d) Reordered coefficients after applying modified zigzag scan (For simplicity, we have omitted quantization step)

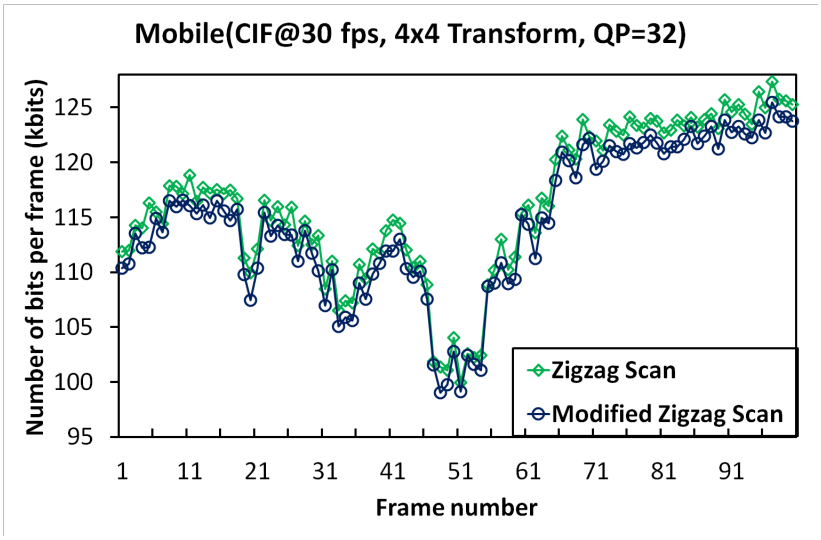


Figure 4.13: Analysis of output bits per frame for inter-frame coding using DAFL-DCT

where P_{00}, P_{01}, P_{02} and P_{03} represent DC coefficients and P_{iu}, $(u = 1, 2, \cdots, N-1)$ depict high frequency AC coefficients with increasing 'i' index corresponding to 'u' index.

Since 1D-DAFL-DCT transform mode 3 is selected for given MC-residual block, the proposed new modified zigzag scan is applied on transformed and quantized block before entropy encoding. For efficient entropy encoding, the coefficients are arranged in scanning order as shown in Figure 4.12(d). The modified scanning order is designed in order to keep high valued quantized DC and low-frequency coefficients at the beginning of the 1D-array and low valued or zero valued high-frequency coefficients at the end with increasing order of 'i' index of transform coefficients for each row vector.

A comparative analysis of output encoded bits using conventional zigzag scan and modified zigzag scan is shown in Figure 4.13. It is observed that DAFL-DCT coefficients are efficiently encoded and therefore, it yields reduced bit-rate for modified zigzag scanning

Algorithm 4.1 Proposed DAFL-DCT transform mode side information coding

Input: (1) TxModeX : DAFL-DCT transform mode of current block X
(2) TxModeA : DAFL-DCT transform mode block A
(3) TxModeB : DAFL-DCT transform mode block B
Output: (1) infoPredTxModeFlag : 1-bit codeword
(2) infoTransformMode : 3-bit codeword
Method:

$$PredTxMode = \begin{cases} 0 & \text{if blocks A \& B are not available,} \\ TxModeA & \text{if block B is not available,} \\ TxModeB & \text{if block A is not available,} \\ min(TxModeA, TxModeB) & \text{otherwise} \end{cases}$$

$diffTxMode = TxModeX - PredTxMode$
if $diffTxMode == 0$ **then**
  $infoPredTxModeFlag = 1$ ▷ $1 - bit\ codeword$
else
  $infoPredTxModeFlag = 0$
end if
if $infoPredTxModeFlag == 0$ **then**
  if $TxModeX < PredTxMode$ **then**
    $infoTransformMode = TxModeX$ ▷ $3 - bit\ codeword$
  else
    $infoTransformMode = TxModeX - 1$
  end if
end if

pattern than conventional zigzag scan. The most important feature of new modified scanning order is its suitability to all 1-D DAFL-DCT transform modes. This is an additional difference between the proposed DAFL-DCT and other existing directional transforms. The proposed directional transform scheme uses the same conventional zigzag scanning pattern for intra-coding and for conventional 2D-DCT in inter-coding and a modified zigzag scanning pattern is proposed for all 1D-DAFL-DCT transform modes in inter-coding, whereas DDCT [98], DART [99] and 1D-Transform [97] use different scanning patterns for each directional transform mode.

4.4.2 Coding of side information

In H.264/AVC video coding, along with quantized coefficients, a lot of side information data such as macroblock type, intra-predication mode, motion vectors, etc., are also encoded and a decoder uses this information, in order to perform exact inverse operations and reconstruct video frames. Conventionally, H.264/AVC employs one of the two block transforms of size 4×4 and 8×8 and conveys to the decoder by 1-bit codeword. But, the DAFL-DCT chooses an optimum transform mode from the two sets of transform modes; one for each block transform size (4×4 and 8×8). Therefore, additional overhead bits as side information

for each transform mode are sent to decoder for proper reconstruction of encoded blocks. In this section, we propose an efficient way to encode this side information with minimum overhead bit-rate.

As discussed earlier, neighbouring blocks tend to select same transform modes unless there is a sudden change in the region. Based on this observation, we have proposed a new approach for side information coding by exploiting spatial correlation of neighbouring blocks. In our proposed scheme, all selected transform modes are categorized into two groups; 'predicted' and 'non-predicted'. If a block chooses the same DAFL-DCT mode as its one/ both neighbours (A and B as shown in Figure 4.9), the transform mode is denoted as 'predicted' and represented by 1-bit codeword (*infoPredTxModeFlag*) only. However, if a block opts DAFL-DCT mode dissimilar to its neighbours, it falls into into 'non-predicted' group and requires additional 3-bits for codeword (*infoTransformMode*) to represent the selected transform mode in encoder output. The detailed algorithm of the proposed DAFL-DCT transform mode side information coding is described as **Algorithm 4.1**.

In the proposed DAFL-DCT, if 4×4 block transform is used in intra-coding, then all 16 blocks will be individually encoded by different DAFL-DCT transform modes and 16 codewords will be generated. However, in inter-coding, MC-residual 4×4 blocks, inside a 8×8 block, are forced to choose the same transform mode and represented by only one codeword similar to 1D-Transform [97]. It is observed that many low valued MC-residual 4×4 blocks contain no coefficients after transform and quantization. Therefore, transmitting one codeword as side information for each of these 4×4 blocks in a 8×8 block would increase the average side information bit-rate that degrades the compression performance. So, the restriction in selecting different DAFL-DCT transform mode is compromised for the improvement in compression performance. If a macroblock opts 8×8-block transform, then only 4 codewords are required as side information. The proposed scheme efficiently categories the selected transform modes in 'predicted' and 'non-predicted' groups and encode them quite efficiently with very less overhead bits.

4.5 Experimental results and discussion

In order to investigate the efficacy of the proposed DAFL-DCT scheme, various experiments have been conducted on H.264/AVC joint model reference software (version JM18.6). For experiments, the conventional 2D-DCT based encoder is represented by DCT, while DAFL-DCT with high efficiency is denoted as DAFL-HE and DAFL-DCT low complexity version is portrayed as DAFL-LC. However, the default encoding mode of DAFL-DCT is DAFL-HE. In addition, there are two block transform modes according to transform block size of 4×4 or 8×8. An encoder with particular block transform mode is denoted by adding transform block size as postfix to encoder; for example, DAFL-HE_4×4, DAFL-HE_8×8, etc. If no postfix is used, it indicates the encoder performance is obtained from all transform

Table 4.1: Encoder configuration in JM 18.6 reference software of H.264/AVC

Common Parameters	Intra-Coding	Inter-Coding
FrameRate = 30.0	FramesToBeEncoded = 50	FramesToBeEncoded = 100
DisableIntra16x16 = 1	IntraPeriod = 1	IntraPeriod = 0
EnableIPCM = 0	IDRPeriod = 15	IDRPeriod = 0
NumberBFrames = 0	QPISlice = $\{20, 26, 32, 38\}$	QPISlice = 26
SymbolMode = 0		QPPSlice = $\{20, 26, 32, 38\}$
PicInterlace = 0		DisableSubpelME = 0
MbInterlace = 0		SearchRange = 32
RDOptimization = 1		NumberReferenceFrames = 1
YUVFormat = 1		DisableIntraInInter = 1
SourceBitDepthLuma = 8		
SourceBitDepthChroma = 8		
Transform8x8Mode = $\{0, 2\}$		

modes.

4.5.1 Experimental set-up

All experiments are carried out on standard video sequences like *Foreman, Highway* and *Mobile*. Video sequences are categorized in terms of their resolutions as QCIF, CIF, 4CIF and HD 720p. Table 3.5 lists the details of the test video sequences. All sequences have complex motions and are rich in directional featured blocks.

We have considered the following two modes to encode video frames.

- Intra-coding with all intra frames (intra-frame only)
- Inter-coding with frame pattern IPPP$\cdots$ (first I-frame and remaining all P-frames)

In intra-coding, all frames are IP-frames and encoded by DAFL-DCT. In inter-coding, in our experiments we do not encode first I-frame with DAFL-DCT, so that performance of DAFL-DCT can be measured only for inter-frame encoding. For simplicity and to avoid any kind of ambiguity in performance comparison, in intra-coding, available block sizes are 4×4 and/ or 8×8 and similarly, in inter-coding, available block sizes are 8×8, 8×4, 4×8 and 4×4. The video sequences are encoded by a set of four quantization parameter (QP) values 20, 26, 32 and 38. Entropy encoding mode is set to CAVLC mode. The detailed encoder configuration for JM18.6 is listed on Table 4.1.

4.5.2 Experiment 1: Bjontegaard metrics performance

We have considered Bjontegaard delta bit-rate (BD-bitrate), Bjontegaard delta PSNR (BD-PSNR) and Bjontegaard delta SSIM (BD-SSIM) as benchmark metrics to evaluate efficacy of the proposed scheme. Bjontegaard metrics calculate average change in bit-rate or PSNR difference between two encoders' R-D curves [36]. In Bjontegaard metric, positive numbers in BD-PSNR and BD-SSIM represent gain, while negative numbers in BD-bitrate show reduction in bit-rate. The BD-PSNR, BD-SSIM and BD-bitrate comparisons are

Table 4.2: Bjontegaard metric[36] performance for 4×4 block transform in H.264/AVC in CAVLC platform

		Intra-Coding		Inter-Coding	
	Sequence	DAFL-HE	DAFL-LC	DAFL-HE	DAFL-LC
BD-PSNR (in dB)	Foreman	0.37	0.27	0.31	0.05
	Highway	0.29	0.21	0.16	0.06
	Mobile	0.58	0.41	0.47	0.22
	Bus	0.42	0.29	0.31	0.13
	Crew	0.23	0.19	0.20	-0.01
	Soccer	0.31	0.23	0.27	0.13
	Old town cross	0.27	0.20	-0.08	-0.05
	Park joy	0.48	0.33	0.33	0.20
	Average	**0.37**	**0.27**	**0.25**	**0.09**
BD-SSIM	Foreman	0.0027	0.0020	0.0019	0.0003
	Highway	0.0036	0.0028	0.0014	0.0004
	Mobile	0.0055	0.0039	0.0048	0.0022
	Bus	0.0048	0.0033	0.0021	0.0003
	Crew	0.0018	0.0015	0.0011	-0.0007
	Soccer	0.0035	0.0024	0.0006	-0.0002
	Old town cross	0.0033	0.0023	-0.0022	-0.0012
	Park joy	0.0059	0.0041	0.0027	0.0000
	Average	**0.0039**	**0.0028**	**0.0015**	**0.0001**
BD-bitrate (%)	Foreman	-6.21	-4.54	-5.87	-1.14
	Highway	-7.31	-5.26	-7.46	-3.01
	Mobile	-6.60	-4.70	-8.05	-3.89
	Bus	-6.63	-4.76	-5.27	-2.02
	Crew	-4.65	-3.95	-6.01	-0.29
	Soccer	-6.10	-4.58	-7.52	-3.72
	Old town cross	-6.70	-4.89	-13.51	-5.81
	Park joy	-7.53	-5.29	-5.32	-3.11
	Average	**-6.47**	**-4.75**	**-7.38**	**-2.88**

summarised in Table 4.2 and Table 4.3 for 4×4 and 8×8 block transform, respectively. The R-D curves of *Mobile* and *Park joy* video sequences are shown in Figure 4.14 and Figure 4.15 that exhibit performance comparisons between conventional 2D-DCT and proposed DAFL-DCT for intra-coding and inter-coding, respectively. In general, following observations are made.

1. In intra-coding, significant reductions in BD-bitrate are observed for the proposed DAFL-HE and DAFL-LC. DAFL-HE achieves improvement in BD-PSNR of 0.37 dB (or equivalently 6.47% reduction in BD-bitrate) and 0.26 dB (or equivalently 5.30% reduction in BD-bitrate) on average for 4×4 and 8×8 block transforms, respectively. It is also noticed that the proposed DAFL-HE with 4×4 block transform achieves BD-PSNR of 0.58 dB (or equivalently 6.60% reduction in BD-bitrate) for *Mobile* and 0.48 dB (or equivalently 7.53% reduction in BD-bitrate) for *Park joy* video sequences. It is observed that video sequences having less directional featured blocks yield lower improvements in BD-PSNR (or equivalently in BD-bitrate savings) as these non-oriented blocks prefer 2D-DCT for encoding and give a little space

Table 4.3: Bjontegaard metric[36] performance for 8×8 block transform in H.264/AVC in CAVLC platform

		Intra-Coding		Inter-Coding	
	Sequence	DAFL-HE	DAFL-LC	DAFL-HE	DAFL-LC
BD-PSNR (in dB)	Foreman	0.32	0.21	0.53	0.16
	Highway	0.22	0.16	0.35	0.25
	Mobile	0.38	0.25	0.74	0.41
	Bus	0.29	0.21	0.40	0.23
	Crew	0.21	0.16	0.34	0.06
	Soccer	0.23	0.17	0.44	0.23
	Old town cross	0.19	0.14	0.22	0.10
	Park joy	0.28	0.21	0.49	0.29
	Average	**0.26**	**0.19**	**0.44**	**0.22**
BD-SSIM	Foreman	0.0029	0.0019	0.0037	0.0009
	Highway	0.0032	0.0024	0.0034	0.0022
	Mobile	0.0043	0.0031	0.0075	0.0040
	Bus	0.0039	0.0027	0.0031	0.0014
	Crew	0.0020	0.0014	0.0019	-0.0010
	Soccer	0.0030	0.0021	0.0020	0.0005
	Old town cross	0.0025	0.0018	0.0019	0.0007
	Park joy	0.0040	0.0030	0.0067	0.0023
	Average	**0.0032**	**0.0023**	**0.0038**	**0.0014**
BD-bitrate (%)	Foreman	-5.88	-3.84	-9.94	-3.02
	Highway	-7.08	-5.32	-13.75	-10.03
	Mobile	-4.47	-3.02	-11.65	-6.73
	Bus	-4.82	-3.47	-5.66	-3.32
	Crew	-5.21	-3.98	-9.67	-2.12
	Soccer	-5.01	-3.82	-10.86	-5.68
	Old town cross	-5.15	-3.75	-16.24	-7.29
	Park joy	-4.79	-3.70	-7.43	-4.62
	Average	**-5.30**	**-3.86**	**-10.65**	**-5.35**

for improvement by the proposed scheme. Moreover, positive values of BD-SSIM indicate superior visual quality of the proposed DAFL-DCT scheme.

2. In inter-coding, DAFL-HE shows improvement in BD-PSNR of 0.25 dB (or equivalently 7.38% reduction in BD-bitrate) and 0.44 dB (or equivalently 10.65% reduction in BD-bitrate) on average for 4×4 and 8×8 block transforms, respectively. It is also found that the proposed DAFL-HE with 8×8 block transform achieves a quite noticeable BD-PSNR of 0.74 dB (or equivalently 11.65% reduction in BD-bitrate) for *Mobile* and 0.49 dB (or equivalently 7.43% reduction in BD-bitrate) for *Park joy* video sequences. For high and complex motion video sequences (*Highway, Mobile, Bus, Soccer and Park joy*) it gives superior compression performance as compared to low and simple motion video sequences like *Foreman and Crew*.

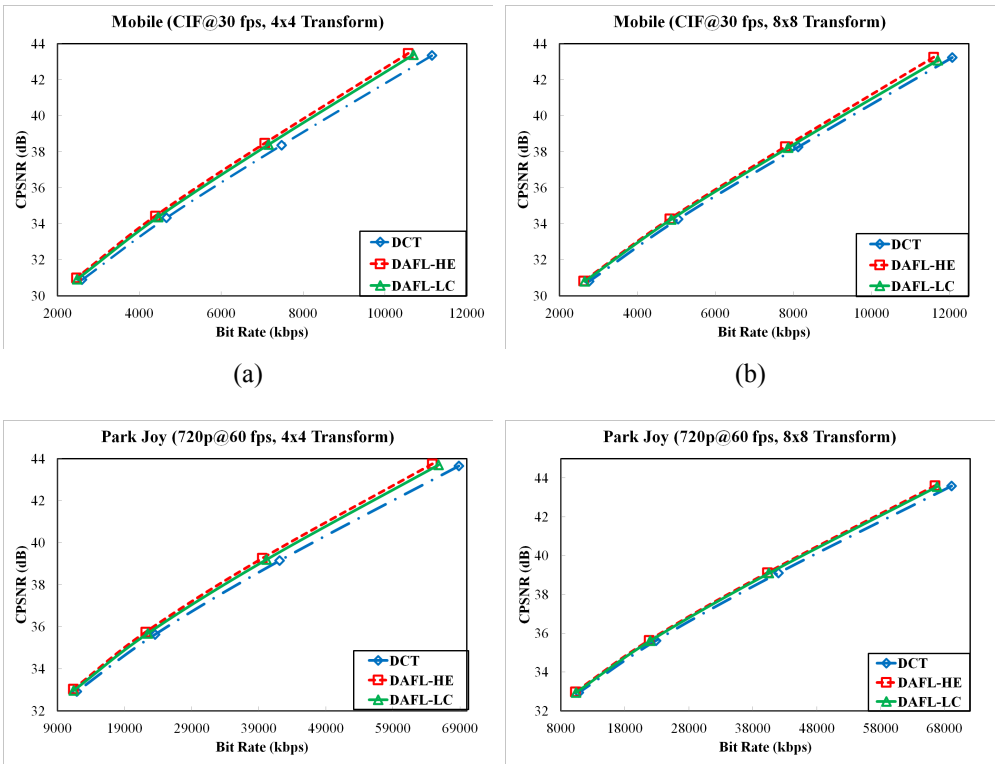


Figure 4.14: Rate-distortion curves for intra coding for *Mobile* and *Park joy* sequences

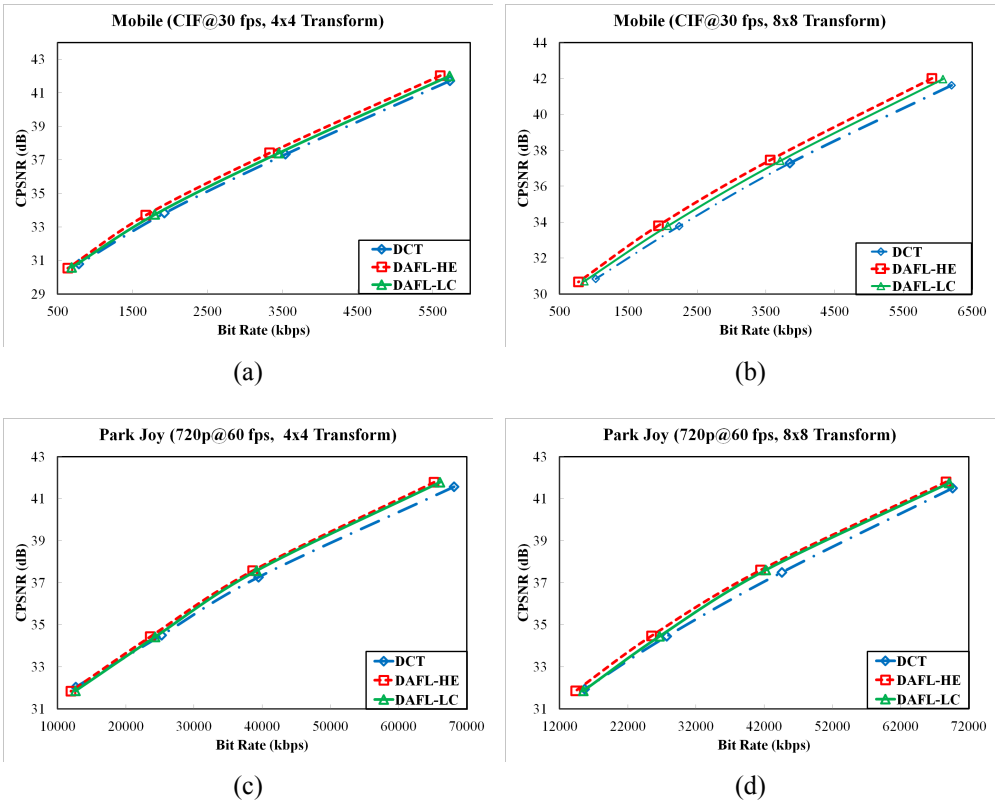


Figure 4.15: Rate-distortion curves for inter coding for *Mobile* and *Park joy* sequences

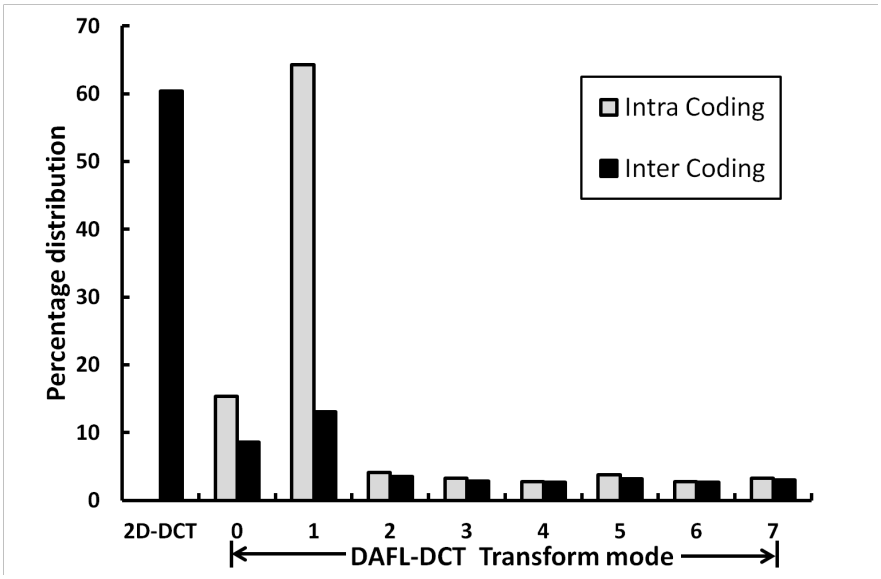


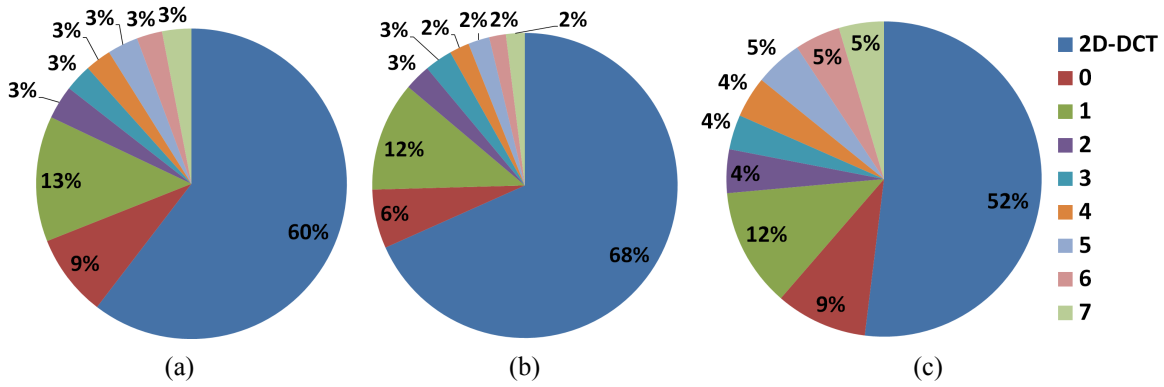
Figure 4.16: Overall percentage distribution of DAFL-DCT transform modes for intra-, and inter-coding

4.5.3 Experiment 2: Transform mode selection

In this experiment, we have investigated that how many number of times each transform mode is selected at various coding modes. The overall percentage distribution of intra-, and inter-coding is shown in Figure 4.16. The percentages are obtained from all video sequences for all block transform modes and for all QP values. The percentage distribution reflects that 2D-DCT (inter-frame coding) and DAFL-DCT transform mode 1 (intra-frame coding) are most preferred transform modes (around 60%) than the other transform modes. In a typical video frame, 40% directional featured blocks are reasonable numbers, as all blocks do not have governing directional features. It is also worth mentioning that at higher bit-rates (lower QP value), the percentage of directional transform mode selection is high at around 45.10% than approximately 33.13% at lower bit-rates. In Figure 4.17, the average percentage distribution of transform modes in inter-coding of *Foreman* video sequence at lower resolution and of *Park joy* video sequence at higher resolution are shown along-with overall average percentage distribution. It is observed that at lower resolution 2D-DCT is opted more frequently; whereas at higher resolutions, 1D-DAFL-DCT modes are mostly preferred.

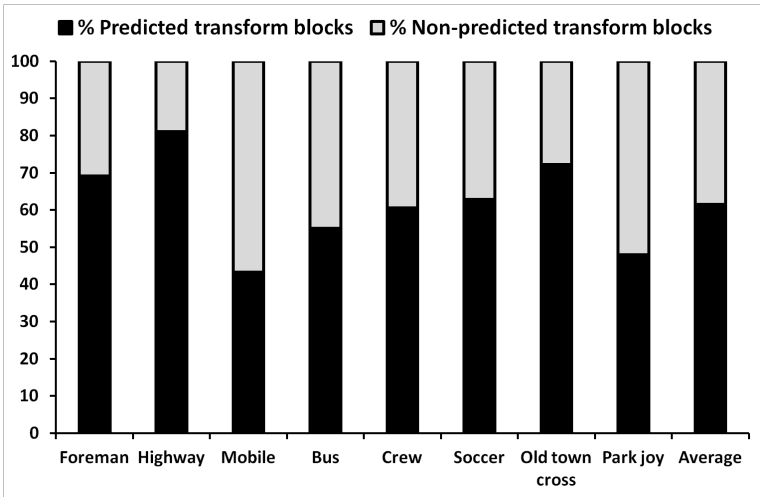
4.5.4 Experiment 3: Side information

In the proposed DAFL-DCT scheme, side information represents the overhead bits needed to transmit to a decoder to inform the selected transform mode out of all available directional transform modes. In our proposed scheme, we have introduced an efficient technique to generate side information with minimum bit-rate overhead. As discussed earlier, transform modes of all blocks are categorised in two groups: predicted and non-predicted. Further,

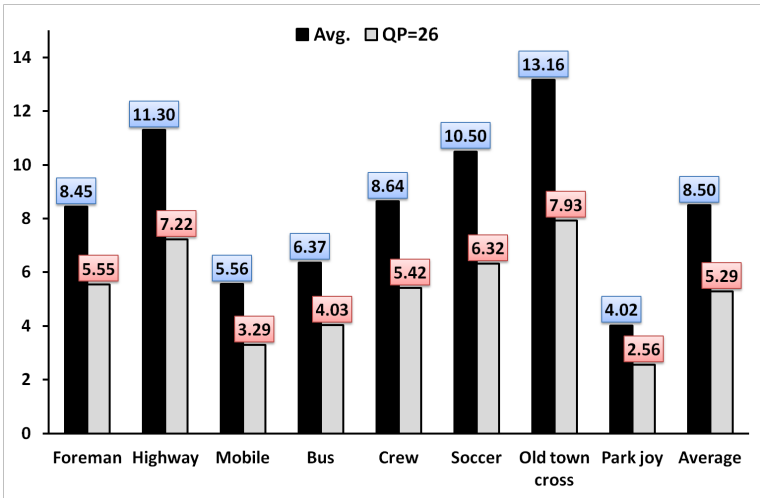


(a) (b) (c)

Figure 4.17: Percentage distribution of DAFL-DCT transform modes in inter-coding. (a) *average*, (b) *Foreman* and (c) *Park joy*



(a)



(b)

Figure 4.18: Side information distribution: (a) distribution of 'predicted' and 'non-predicted' transform blocks, (b) percentage of total bit-rate used in side information

due to spatial redundancy, neighbours have tendency to opt similar transform modes. This characteristic is exploited to minimize side information bit-rate. Each predicted block needs

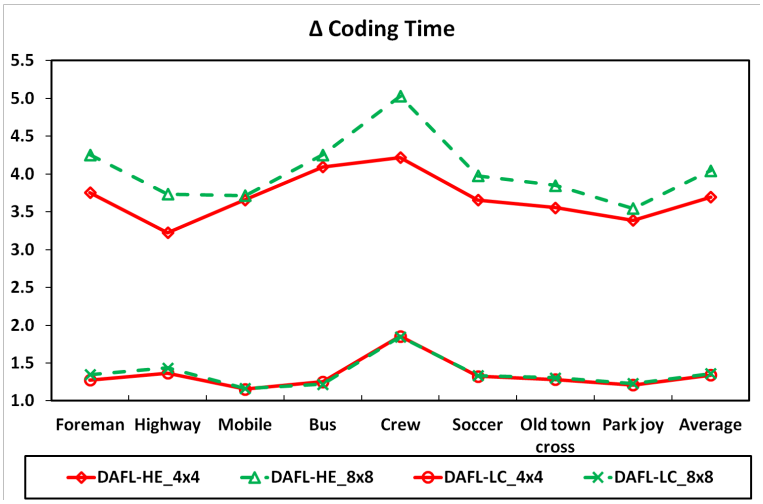
1-bit codeword and every non-predicted block requires 4-bit codeword to inform selected transform.

The percentage distribution of predicted and non-predicted transform blocks for each video sequence in inter-coding is shown in Figure 4.18(a). The average numbers are obtained from all encoding options. The average percentage of predicted and non-predicted blocks are 62.64% and 37.36%, respectively. The high motion and complex video sequences (*Mobile, Bus* and *Park joy*) have less predicted transform blocks as compared to low motion and simple video sequences. Figure 4.18(b) presents percentage of total bit-rates used in side information on average and for higher bit-rate (QP=26) considering all available block transform modes for all video sequences in inter-coding. The average percentage of bit-rate used for side information is 8.50%, while for QP at 26, the average percentage is only 5.29%.

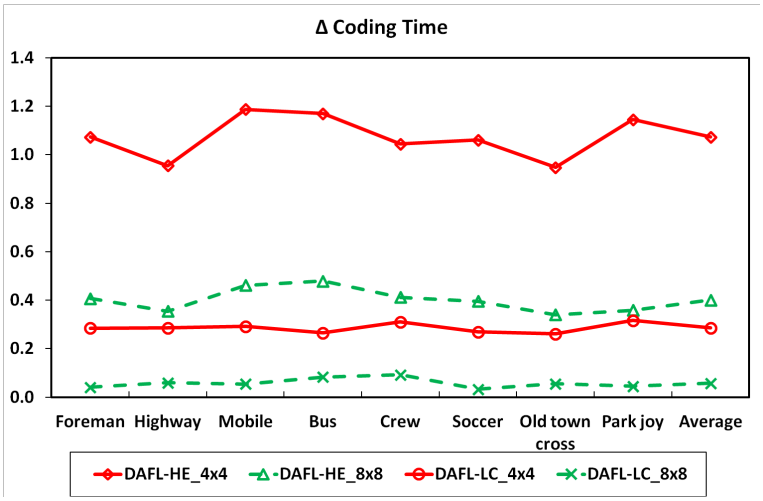
4.5.5 Experiment 4: Analysis of encoding time complexity

In the proposed DAFL-DCT scheme, an optimum transform mode is selected for each block from all available transform modes (8 for intra-, and 9 for inter-coding) using RDO. RDO based selection process not only achieves higher compression performance, but also increases computational overhead. In RDO, each block has to go through various encoding modules such as transformation, quantization, entropy encoding and then inverse operations for every available directional transform mode in sequential order. Moreover, DAFL-DCT uses directional transpose and its inverse to rearrange block data that costs as extra encoding time. Hence, encoding time increases considerably with number of available transform modes. The proposed DAFL-HE chooses an optimum transform mode from globally available transform modes. On the other hand, DAFL-LC selects from a variable length local set of transform modes that reduces its encoding time significantly.

In this experiment, in order to compute the encoding time complexity, we have calculated encoding time for each candidate encoder: conventional 2D-DCT, DAFL-HE and DAFL-LC for intra-, and inter-coding. The relative change in encoding time ΔT is calculated by (3.20) , where encoding time for the conventional 2D-DCT is considered as reference. The positive numbers represent increase in coding time with respect to conventional 2D-DCT and vice-versa. Figure 4.19(a) and 4.19(b) represent ΔT for intra-, and inter-coding, respectively. It is observed from Figure 4.19(a) that encoding time varies significantly for different block transforms. The average ΔT of DAFL-HE for 4×4 block transform mode is equal to 3.6 whereas it is 4.0 for 8×8 block transform. In addition, ΔT performance of DAFL-LC is also quite promising. The average ΔT of DAFL-LC is equal to 1.3 and 1.4 for 4×4 and 8×8 block transforms, respectively. In inter-coding, as shown in Figure 4.19(b), the encoding complexity is almost negligible, as it employs 1D-DAFL-DCTs that require considerably less time than 2D-DAFL-DCT. The ΔT is equal to 1.0 for 4×4 block transform whereas 0.4 only for 8×8 block transform in DAFL-HE. In DAFL-LC, the ΔT is equal to 0.3 for 4×4 block transform whereas 0.1 only for 8×8 block transform on average by considering



(a)



(b)

Figure 4.19: Δ Coding time: (a) intra-coding, (b) inter-coding

all QP values of all video sequences. Hence, it may be concluded that the proposed scheme has introduced marginal increment in encoding time-complexity but has achieved superior compression performance for directional featured blocks.

4.5.6 Experiment 5: Subjective performance

The subjective performance comparisons of DAFL-DCT against conventional 2D-DCT are shown in Figure 4.20 for enlarged version of a portion of *Mobile* sequence. It is clearly seen that the spokes of wheel, the leaves and the object boundaries are visually more appealing as compared to those produced by its competitive scheme. It is noticed that directional edges are more sharp and clean in case of DAFL-DCT.

(a) (b)

(c) (d)

Figure 4.20: Subjective performance of DAFL-DCT and conventional 2D-DCT for *Mobile* sequence coded with 8×8 block transform mode (only a portion a shown here). a) reconstructed 10^{th} I-frame (1777.82 kbps, 28.31 dB) by DCT, b) reconstructed 10^{th} I-frame (1745.24 kbps, 28.78 dB) by DAFL-DCT, c) reconstructed 45^{th} P-frame ($764.11.50$ kbps, 30.01 dB) by DCT and d) reconstructed 45^{th} P-frame (760.4 kbps, 30.85 dB) by DAFL-DCT

4.5.7 Experiment 6: Comparison with other directional transforms

To further analyse the performance of the proposed DAFL-DCT scheme, it has been compared against other existing directional transforms for context adaptive binary arithmetic coding (CABAC) entropy coding mode. The other directional transforms include DDCT [98], DART [99] and 1D-Transform [97]. In these directional transforms, both DDCT and DART are proposed for both intra-, and inter-coding, whereas 1D-Transform is proposed for inter-coding only. All directional transform modes of DDCT and DART are implemented for 4×4 and 8×8 blocks. In 1D-Transform, all transform modes for 4×4 blocks are applied, but only eight directional transform modes which appear similar to transform modes of DAFL-DCT have been implemented for 8×8 blocks out of sixteen transform modes. The BD-PSNR and BD-bitrate comparisons of these transforms against conventional 2D-DCT are summarized on Tables 4.4 and 4.5 for 4×4 and 8×8 block transforms, respectively.

In intra-coding as observed, DDCT yields improvement in BD-PSNR of 0.62 dB (or equivalently 8.38% reduction in BD-bitrate) and BD-PSNR of 0.20 dB (or equivalently

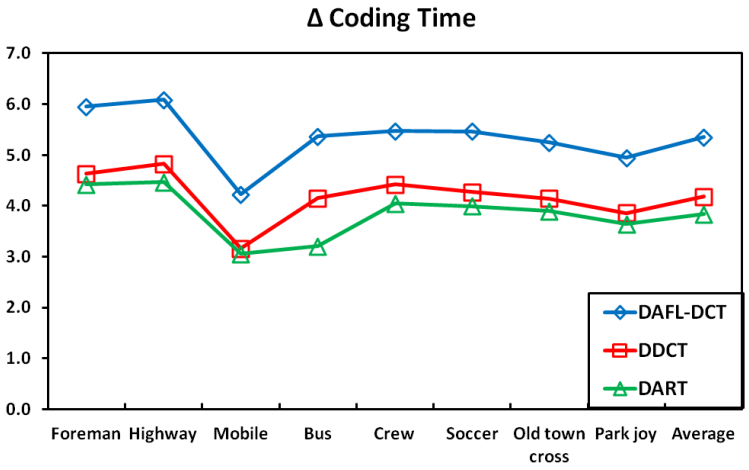
Table 4.4: Bjontegaard metric[36] performance comparison of other directional transforms for 4×4 block transform in H.264/AVC in CABAC platform

	Sequence	Intra-Coding			Inter-Coding			
		DAFL-DCT	DDCT [98]	DART [99]	DAFL-DCT	1D-Transform [97]	DDCT [98]	DART [99]
BD-PSNR (dB)	Foreman	0.65	**0.69**	0.52	**0.21**	0.04	0.18	-0.03
	Highway	0.47	0.49	**0.58**	**0.26**	0.20	0.15	0.10
	Mobile	**0.91**	0.88	0.84	**0.68**	0.37	0.45	0.41
	Bus	**0.83**	0.75	0.48	**0.43**	0.17	0.24	0.03
	Crew	**0.46**	0.38	0.26	**0.10**	0.03	0.07	0.05
	Soccer	**0.45**	0.36	0.15	**0.19**	0.05	0.12	0.04
	Old town cross	**0.67**	0.58	0.48	**0.21**	0.09	0.07	0.08
	Park joy	**0.90**	0.84	0.65	**0.43**	0.13	0.30	0.14
	Average	**0.67**	**0.62**	**0.49**	**0.31**	**0.13**	**0.20**	**0.10**
BD-bitrate (%)	Foreman	-8.37	**-8.93**	-6.71	**-3.10**	-0.49	-2.87	0.50
	Highway	-8.46	-8.66	**-10.20**	**-7.74**	-6.79	-4.22	-3.43
	Mobile	**-8.00**	-7.82	-7.22	**-8.85**	-4.50	-6.09	-5.44
	Bus	**-9.22**	-8.37	-5.47	**-6.11**	-2.53	-3.80	-0.44
	Crew	**-9.27**	-7.75	-5.26	**-2.11**	-1.04	-1.47	-1.20
	Soccer	**-6.78**	-5.52	-2.39	**-3.86**	-1.03	-2.68	-0.98
	Old town cross	**-11.84**	-10.38	-8.39	**-8.49**	-4.64	-1.91	-4.40
	Park joy	**-10.21**	-9.57	-7.32	**-5.79**	-1.95	-4.09	-1.75
	Average	**-9.02**	**-8.38**	**-6.62**	**-5.76**	**-2.87**	**-3.39**	**-2.14**

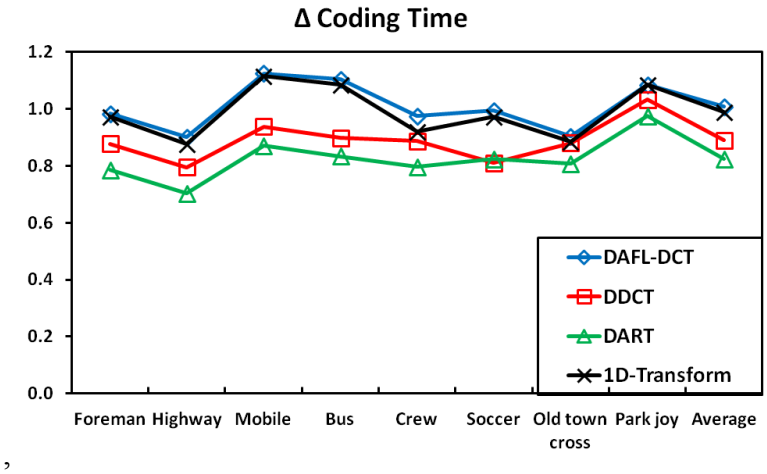
Table 4.5: Bjontegaard metric[36] performance comparison of other directional transforms for 8×8 block transform in H.264/AVC in CABAC platform

	Sequence	Intra-Coding			Inter-Coding			
		DAFL-DCT	DDCT [98]	DART [99]	DAFL-DCT	1D-Transform [97]	DDCT [98]	DART [99]
BD-PSNR (dB)	Foreman	0.35	**0.38**	0.00	**0.45**	0.39	0.38	0.03
	Highway	0.29	**0.30**	0.30	**0.73**	0.66	0.35	0.54
	Mobile	**0.37**	0.21	0.32	**0.99**	0.81	0.51	0.59
	Bus	**0.34**	0.15	-0.03	**0.69**	0.56	0.45	0.05
	Crew	**0.25**	0.15	0.01	**0.20**	0.06	0.14	0.00
	Soccer	**0.20**	0.02	0.04	**0.37**	0.27	0.29	0.11
	Old town cross	**0.26**	0.15	0.09	**0.50**	0.33	0.39	0.26
	Park joy	**0.34**	0.25	0.11	**0.66**	0.53	0.25	0.26
	Average	**0.30**	**0.20**	**0.10**	**0.57**	**0.45**	**0.35**	**0.23**
BD-bitrate (%)	Foreman	-5.13	**-5.89**	-0.10	**-6.91**	-6.04	-6.17	-0.29
	Highway	-5.88	**-6.13**	-6.12	**-18.08**	-16.66	-9.41	-13.83
	Mobile	**-3.41**	-2.00	-2.79	**-12.90**	-10.57	-6.87	-7.80
	Bus	**-4.07**	-1.82	0.35	**-9.19**	-7.49	-6.26	-0.60
	Crew	**-5.77**	-3.60	-0.18	**-4.26**	-1.50	-3.14	-0.12
	Soccer	**-3.31**	-0.43	-0.53	**-7.36**	-5.39	-5.99	-1.96
	Old town cross	**-5.24**	-3.10	-1.61	**-17.90**	-12.56	-14.94	-9.67
	Park joy	**-4.17**	-3.18	-1.42	**-7.61**	-6.40	-3.66	-2.90
	Average	**-4.62**	**-3.27**	**-1.55**	**-10.53**	**-8.33**	**-7.06**	**-4.65**

3.27% reduction in BD-bitrate) on average; the DART results in improvement in BD-PSNR of 0.49 dB (or equivalently 6.62% reduction in BD-bitrate) and BD-PSNR of 0.10 dB (or equivalently 1.55% reduction in BD-bitrate) on average; whereas, the proposed DAFL-DCT achieves superior performance, improvement in BD-PSNR of 0.67 dB (or equivalently



(a)



(b)

Figure 4.21: Comparison of Δ coding time of the proposed DAFL-DCT against existing directional transforms for 4×4 block transform: a) intra-coding, b) inter-coding

9.02% reduction in BD-bitrate) and BD-PSNR of 0.30 dB (or equivalently 4.62% reduction in BD-bitrate) on average for 4×4 and 8×8 block transforms, respectively.

In inter-coding as observed, improvement in BD-PSNR in 1D-Transform of 0.13 dB (or equivalently 2.87% reduction in BD-bitrate) and 0.45 dB (or equivalently 8.33% reduction in BD-bitrate), DDCT of 0.20 dB (or equivalently 3.39% reduction in BD-bitrate) and 0.35 dB (or equivalently 7.06% reduction in BD-bitrate), DART of 0.10 dB (or equivalently 2.14% reduction in BD-bitrate) and 0.23 dB (or equivalently 4.65% reduction in BD-bitrate) and the proposed DAFL-DCT of 0.31 dB (or equivalently 5.76% reduction in BD-bitrate) and 0.57 dB (or equivalently 10.53% reduction in BD-bitrate) on average for 4×4 and 8×8 block transforms, respectively.

Here, we have also presented a comparative analysis of ΔT for theses existing directional transforms along with the proposed DAFL-DCT (DAFL-DCT's default mode is DAFL-HE) against conventional DCT. The ΔT for intra-, and inter-coding for 4×4 block are shown

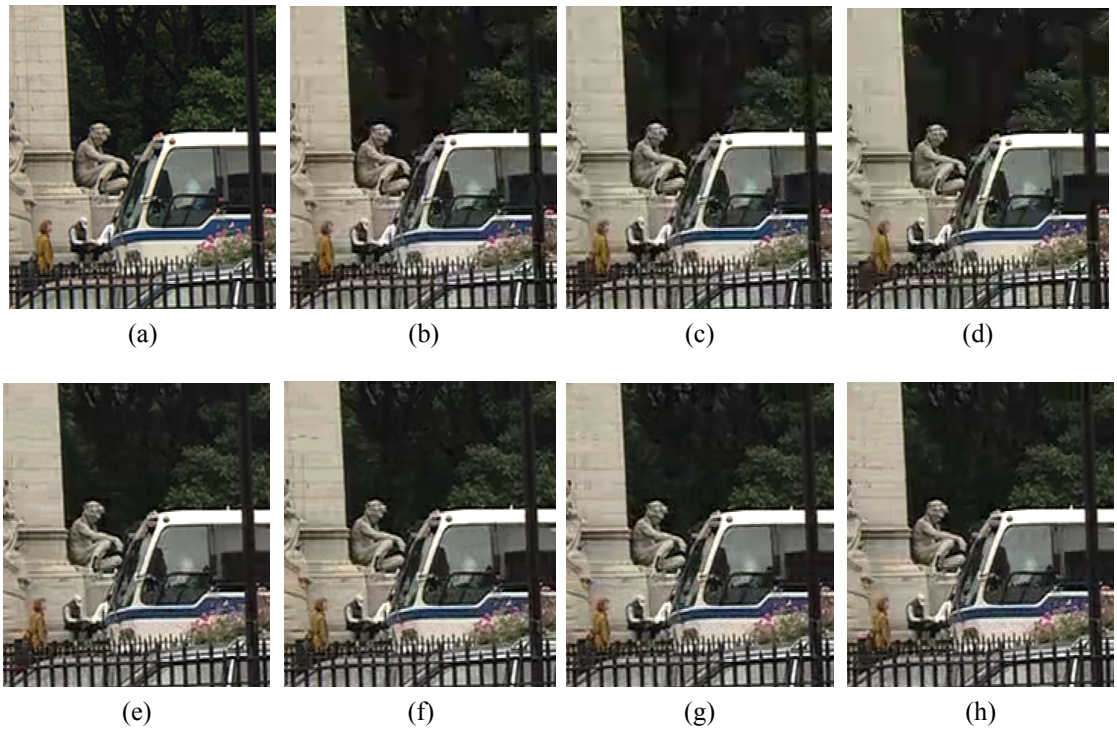


Figure 4.22: Subjective performance of DAFL-DCT and other existing state-of-the-art directional transforms for *Bus* sequence 13^{th}-frame coded with 8×8 block transform mode (only a portion a shown here). a) Original frame b) reconstructed I-frame (1994.78 kbps, 31.32 dB) by DAFL-DCT, c) reconstructed I-frame (1995.24 kbps, 30.57 dB) by DDCT and d) reconstructed I-frame (2036.58 kbps, 29.54 dB) by DART, e)reconstructed P-frame (933.18 kbps, 31.91 dB) by 1D-Transform, f) reconstructed P-frame (915.19 kbps, 32.19 dB) by DAFL-DCT, g) reconstructed P-frame (920.24 kbps, 32.01 dB) by DDCT and h) reconstructed P-frame (946.25 kbps, 31.78 dB) by DART

in Figure 4.21, respectively. It can be observed that in intra-coding, ΔT of DAFL-DCT is slightly higher as compared to other directional transforms. However, it is also observed that ΔT of DAFL-DCT is almost same as for other directional transforms for inter-coding. Unlike the other existing directional transforms where DCT is directly applied to the pixels, in our proposed DAFL-DCT, pixels are first rearranged and then DCT is applied to those blocks. This adds up to the additional time complexity of the DAFL-DCT encoder, whereas the compression efficiency of the proposed DAFL-DCT is improved. So, the increase in encoding time complexity is compromised to some extent for the improvement in efficiency of the DAFL-DCT encoder.

The subjective performance of the proposed DAFL-DCT is also compared against these directional transforms as shown in Figure 4.22 for a cropped version of *Bus* sequence. It can be seen that lines on walls, leaves and curves of the statue are visually prominent in reconstructed frames by DAFL-DCT than other directional transforms for both intra-, and inter-coding.

4.6 Conclusion

In this chapter, we have proposed an efficient direction-adaptive fixed length discrete cosine transform (DAFL-DCT) for directional featured blocks. The DAFL-DCT proposes two sets of directional transform modes for 4×4 and 8×8 blocks, one for each. The proposed scheme takes 4×4 or 8×8 blocks and rearrange to new coordinates based on selected directional transform mode. Later, DCT is performed on these directionally transposed blocks to exploit directional spatial correlation among pixels. Fixed length directional DCTs, having easy implementation and less computational cost, make the proposed DAFL-DCT a suitable candidate for directional featured block transform in real-time applications. In this chapter, we have also proposed a low complexity mode of DAFL-DCT, a new modified zigzag scanning pattern for 1D-DAFL-DCTs in inter-frame coding and an efficient side information coding scheme for DAFL-DCT transform modes. These features have significantly improved the performance of the proposed DAFL-DCT. The proposed DAFL-DCT is shown to have superior performance than the conventional 2D-DCT and other existing directional transforms in terms of both the quantitative and qualitative analysis. The proposed directional transform scheme with the introduction of new efficient motion estimation schemes, to remove temporal redundancy, is expect to lead to further improvement in compression performance in rich media applications.

Chapter 5

Development of Fast Motion Estimation Schemes

Preview

Motion estimation (ME) is employed in video compression schemes to reduce temporal correlation among video frames and yield significant improvement in compression ratio. Among various ME schemes, block-matching motion estimation (BMME) is the most popular approach due to its simplicity and efficiency. The real-world video sequences may contain slow, medium and/ or fast motion activities. Further, a single search pattern does not prove efficient in finding best matched block for all motion types. In this chapter, an efficient direction-adaptive motion estimation (DAME) scheme is proposed which adaptively selects shape and size of the patterns based on motion content. In addition, it is observed that most of the BMME schemes are based on uni-modal error surface. Nevertheless, real-world video sequences may have many local minima available within a search window and thus possess multi-modal error surface. To resolve the local minima problem, we also propose a pattern-based modified particle swarm optimization motion estimation (PMPSO-ME) scheme. Performance analysis of DAME and PMPSO-ME schemes on JM 18.6 of H.264/AVC platform reveals that the proposed schemes outperform other existing BMME schemes by yielding lower computational complexity without degrading visual quality.

The following topics are covered in this chapter.

- Introduction
- Fundamentals of motion estimation
- Development of direction-adaptive motion estimation (DAME) scheme
- Development of pattern-based modified particle swarm optimization motion estimation (PMPSO-ME) scheme
- Experimental results and discussion
- Conclusion

5.1 Introduction

Motion estimation (ME) exploits temporal correlation among video frames and yields significant improvement in compression ratio while sustaining high visual quality in video coding. Block-matching motion estimation (BMME) scheme determines the best match for the current block in a reference frame and yields displacement of the block in terms of motion vector (MV).

ME is based on two error surface modes: (1) uni-modal error surface (UES) and (2) multi-modal error surface (MES). In this chapter, we propose two efficient and fast BMME schemes: direction-adaptive motion estimation (DAME) scheme and pattern-based modified particle swarm optimization motion estimation (PMPSO-ME) scheme, one for each error surface model. In DAME scheme, we categorize motion types present on various video sequences into broadly three different types: slow, medium and fast. Since a single search pattern cannot match multiple motion types present in a video sequence, the proposed scheme uses different combinations of search patterns based on motion classification of a block. The search patterns include small diamond search pattern (SDSP), kite search pattern (KSP), cross search pattern (CSP) and hexagonal search pattern. Hexagonal search pattern is further modified and divided into two types of directional hexagonal search patterns: horizontal hexagonal search pattern (HHSP) and vertical hexagonal search pattern (VHSP). With the help of KSP and directional hexagonal search patterns, the proposed scheme achieves directionality and significantly reduces the number of search points and thus speeds up the matching process by successive minimization in UES.

For MES, we have proposed a fast BMME scheme based on modified particle swarm optimization (PSO). PSO is a population-based evolutionary method [144]. It minimizes the chance of getting trapped into local minima. It is observed that PSO based ME schemes are either very slow due to high computational cost or accuracy of estimating MV is compromised to achieve lower complexity [152, 154]. In this chapter, we have proposed pattern-based modified PSO motion estimation (PMPSO-ME) scheme. The PMPSO-ME is low in computational complexity, as it reduces the number of search points significantly without compromising visual quality. Actually, in conventional PSO (CPSO), the accuracy of true MV depends not only on the population size, but also on number of iterations [144]. On the other hand, PMPSO-ME uses some efficient early termination techniques along-with number of iterations to reduce the computational cost. The proposed PMPSO-ME outperforms other existing PSO based ME schemes present in literature and yields superior compression performance.

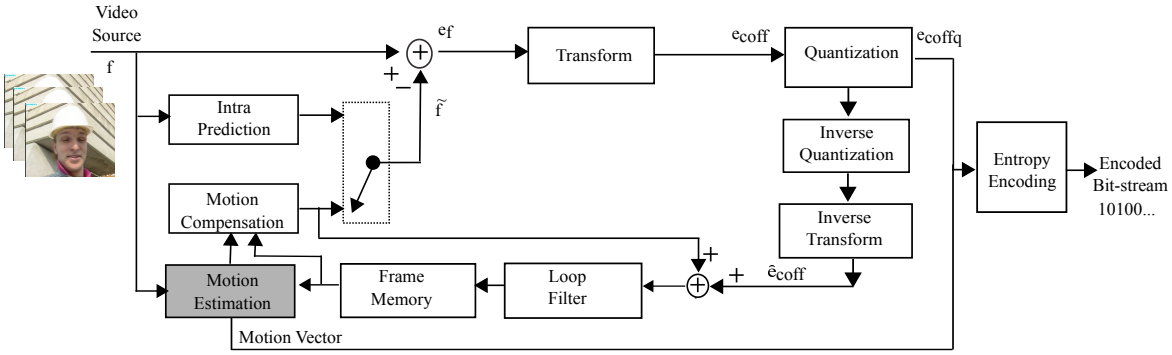


Figure 5.1: Block diagram of H.264/AVC video encoder

5.2 Fundamentals of Motion Estimation

A video sequence consists of video frames. Each frame is encoded either by intra-coding or inter-coding. In intra-coding, each frame is encoded without any reference to other frames. But, a video sequence also contains temporal correlation among its frames. Inter-coding exploits temporal redundancy among frames and yields higher compression performance. In inter-coding, ME finds the best matched block and the motion compensation module generates motion compensated (MC)-residual blocks. The MC-residual blocks are encoded and sent to output bit-stream. The block diagram of H.264/AVC video encoder is illustrated in Figure 5.1.

The detail of BMME process is shown in Figure 5.2. The BMME process searches the best matched block in reference frame ($f_{t-\delta}$) corresponds to the current block (E) present in current frame (f_t) where δ represents time index in temporal domain. The parameter, δ may take on any value from $\{1, 2, \cdots, 16\}$ in H.264/AVC. The displacement of the best matched block from the co-located block is given by MV. For each block of current frame, best matched block is searched in the reference frame inside the search window of size W_s, centred to co-located block. The search window is decided by the maximum displacement

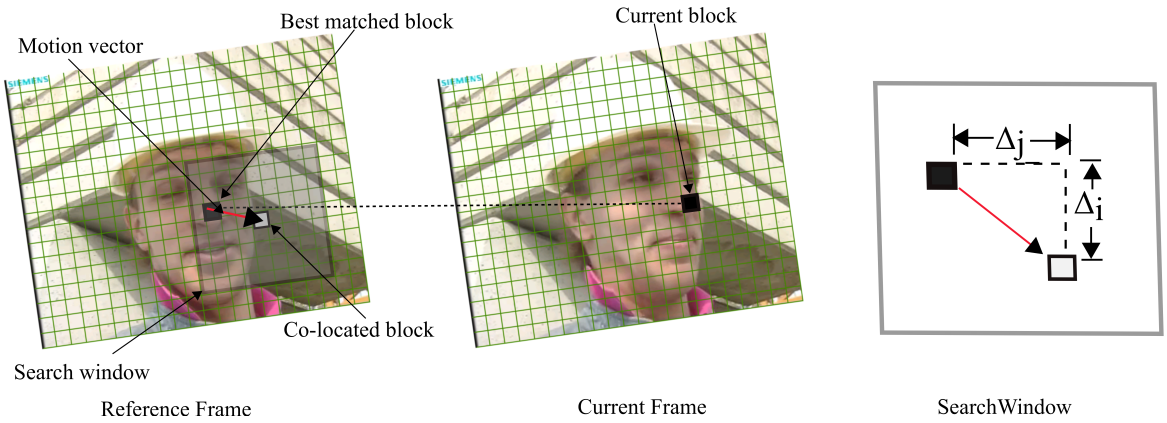


Figure 5.2: Motion estimation (ME) technique

range. MV ($\overrightarrow{\mathrm{mv}}$) is expressed as:

$$\overrightarrow{\mathrm{mv}} = \{\Delta_i, \Delta_j\}' \tag{5.1}$$

where Δ_i and Δ_j represent vertical and horizontal component of $\overrightarrow{\mathrm{mv}}$, respectively.

In process of searching the best matched block, the prediction error is minimized and measured by estimation criteria or distortion metric. Commonly used estimation criteria are sum of absolute difference (SAD), sum of absolute transformed difference (SATD) and mean of squared error (MSE). Among these, SAD is mostly used because it is computationally simple [125, 126, 130]. The proposed DAME scheme also uses SAD as estimation criteria. In general, BMME process can be mathematically expressed as:

$$\overrightarrow{\mathrm{mv_i}} = \underset{\overrightarrow{\mathrm{mv_s}} \in W_s}{\arg\min} \{SAD(\overrightarrow{\mathrm{mv_s}})\} \tag{5.2}$$

where $\overrightarrow{\mathrm{i}} = \{i, j\}'$ represents current block co-ordinates, W_s is the size of search window to which $\overrightarrow{\mathrm{mv_i}}$ belongs and $\overrightarrow{\mathrm{mv_s}}$ depicts all MVs within the search range of $\pm W_s$.

SAD is calculated as:

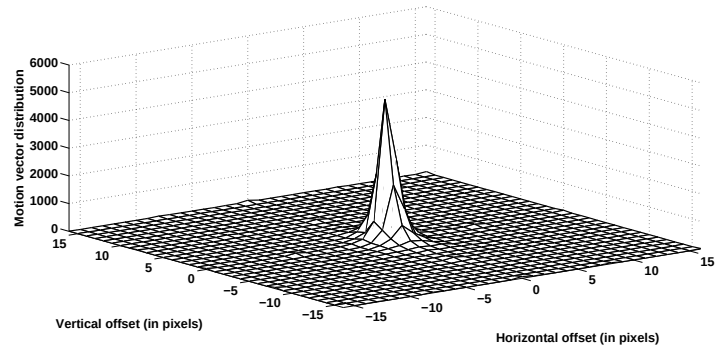
$$SAD(\overrightarrow{\mathrm{mv_i}}) = \sum_{\overrightarrow{\mathrm{k}} \in \mathrm{B_i}} \mid f_t(\overrightarrow{\mathrm{k}}) - f_{t-\delta}(\overrightarrow{\mathrm{k}} - \overrightarrow{\mathrm{mv_i}}) \mid \tag{5.3}$$

where B represents current block of size $M \times N$ and $\overrightarrow{\mathrm{k}}$ is defined as:

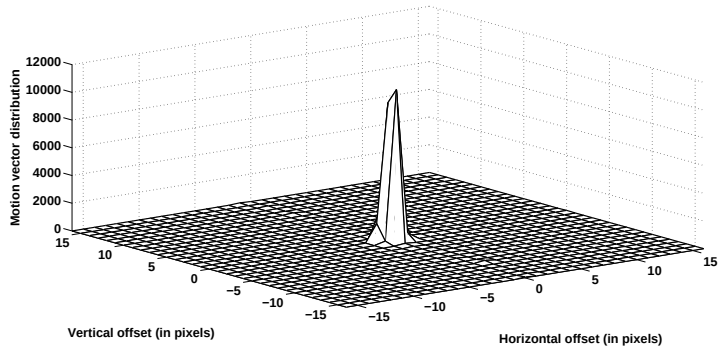
$$\overrightarrow{\mathrm{k}} = \left\{\{k, l\}' : 0 \leq k \leq M-1, 0 \leq l \leq N-1\right\} \tag{5.4}$$

5.3 Development of Direction-Adaptive Motion Estimation (DAME) Scheme

The proposed DAME scheme can be classified under reduced search points based BMME (RSP-BMME), as it checks less number of search points that lead to less number of SAD computations. The proposed DAME scheme exploits spatio-temporal neighbouring blocks' MV correlation characteristics among video frames to predict motion type (slow, medium and fast) of a block. According to the MV distribution of *Foreman* and *Mobile*, shown in Figure 5.3, almost 80% to 90% MVs are of stationary or slow motion type and come inside the central 3×3 area. The detailed MV distribution of different video sequences is presented in Table 5.1.It is observed that, in most of the video sequences, MV distributions are centrally biased. Hence, with the objective of exploiting the characteristics of centre-biased MV distribution, the proposed DAME uses centre-biased search patterns such as SDSP, CSP, KSP, HHSP and VHSP. Most of the search patterns used in literature [130–132, 135, 184, 185] are regular and symmetrical in shape. These search patterns are omnidirectional, i.e., they explore the search points in all directions to determine the true MV



(a) Foreman



(b) Mobile

Figure 5.3: Motion vector distribution with full search method within a search range of ± 16 pixels for 100 frames

and hence are computational intensive. The typical example of UES is illustrated in Figure 5.4 with search window size of 16. Therefore, we propose DAME scheme that employs directional search patterns to improve the searching process. The detailed flowchart of the proposed DAME scheme is shown in the Figure 5.5. Various stages of the proposed scheme are explained below.

Table 5.1: MV distribution based on maximum displacement using FS for search range ± 32

Ψ **(in Pixels)**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8 or more**
Foreman	76.44	13.71	3.76	1.88	1.28	0.79	0.45	0.28	1.11
Highway	90.95	5.09	1.66	0.96	0.59	0.31	0.20	0.08	0.14
Mobile	68.85	20.10	2.97	1.57	0.96	0.45	0.31	0.15	4.49
Bus	65.81	2.87	2.61	2.90	4.39	4.74	3.47	3.61	9.45
Crew	68.74	8.49	4.86	2.71	1.28	0.50	0.33	0.23	12.74
Soccer	73.63	4.07	2.54	4.18	3.51	3.13	2.80	1.98	4.09
Park Joy	56.35	5.33	7.56	3.22	1.40	0.77	0.42	0.17	24.78
Old Town Cross	75.89	12.33	3.50	1.00	0.83	0.61	0.31	0.25	5.29
Average	**72.08**	**9.00**	**3.68**	**2.30**	**1.78**	**1.41**	**1.03**	**0.84**	**7.76**

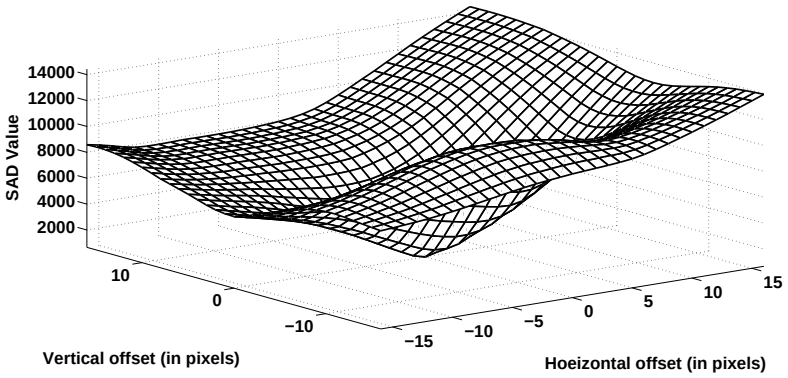


Figure 5.4: Example of uni-modal error surface

Figure 5.5: Flowchart of the proposed directional-adaptive motion estimation (DAME) scheme

5.3.1 Zero motion vector (ZMV) prejudgement

In video sequences, there are many regions in a frame which are stationary such as static object and backgrounds. For these blocks, displacement of the current block is zero. In other words, the co-located blocks in the reference frame are the best matched block and hence, these blocks have zero motion vector (ZMV). ZMV ($\overrightarrow{\mathrm{zmv}}$) is defined as:

$$\overrightarrow{\mathrm{zmv}} = \{0, 0\}' \tag{5.5}$$

Applying search patterns to such blocks increase number of SAD computations unnecessarily and in consequence, a frame encoding time increases. To avoid such instances, the proposed DAME introduces threshold based ZMV detection. If the SAD value of co-located block is less than a threshold value (Υ_{SAD}), then the DAME algorithm terminates immediately; otherwise, it continues.

5.3.2 Selection of motion vector prediction (MVP)

It is observed that MV distribution with respect to MVP has more symmetric shape than ZMV [186]. Here, MVP is predicted by considering spatial neighbouring blocks, left (A), upper (B) and upper right (C) of current frame. In DAME algorithm, spatio-temporal neighbours are used to predict the MV. Therefore, left (X) and upper right (Y) neighbouring blocks and co-located block (Z) of the reference frame are also taken into account. These spatio-temporal neighbouring blocks are illustrated in Figure 5.6. Hence, the DAME uses a total of seven spatio-temporal neighbouring blocks' MV to predict the current MV or MVP. The MVP is calculated as the MV with smallest SAD value. It is expressed as:

$$\overrightarrow{\mathrm{mvp}} = \arg\min_{\overrightarrow{\mathrm{mv}}} \{SAD(\overrightarrow{\mathrm{mvp_1}}), SAD(\overrightarrow{\mathrm{mvp_2}}), SAD(\overrightarrow{\mathrm{zmv}})\} \tag{5.6}$$

where the $\overrightarrow{\mathrm{mvp_1}}$ and $\overrightarrow{\mathrm{mvp_2}}$ are defined as:

$$\overrightarrow{\mathrm{mvp_1}} = median(\overrightarrow{\mathrm{mv_A}}, \overrightarrow{\mathrm{mv_B}}, \overrightarrow{\mathrm{mv_C}}) \tag{5.7}$$

$$\overrightarrow{\mathrm{mvp_2}} = median(\overrightarrow{\mathrm{mv_X}}, \overrightarrow{\mathrm{mv_Y}}, \overrightarrow{\mathrm{mv_Z}}) \tag{5.8}$$

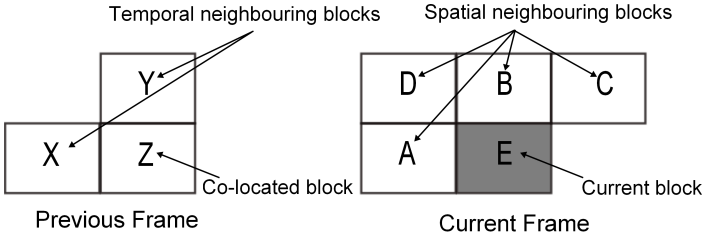


Figure 5.6: Spatio-temporal neighbouring blocks

It is also shown for real-world video sequences that MV distributions are zero-biased or centre-biased with respect to MVP [186]. Hence, the proposed DAME algorithm uses SDSP to check whether the current block is centre-biased or not. If minimum SAD value is at the centre search point of SDSP, then the DAME is terminated immediately; else it continues.

5.3.3 Motion type classification

MV of a block has high correlation to its spatio-temporal neighbours' MV. If the current block and its spatial neighbouring blocks are of same region, then their MV have similar characteristics. However, if the spatial neighbouring blocks belong to different regions or objects, then these blocks have different motion content and hence lead to inaccurate determination of MVP for current block. In such a case, temporal neighbouring blocks play vital role due to temporal correlation unless an abrupt change occurs in the scene.

In the DAME, a motion type (slow/medium/fast) of the current block is defined based on MVP. An appropriate search pattern is selected based on the motion type classification, which not only increases the probability of finding best matched block, but also reduces the encoding time by checking fewer search points. The proposed scheme classifies motion types on the basis of maximum displacement (Ψ) which is defined as the magnitude of the largest component (x- and y-component) of MV. Since the true MV of the current block is unknown, MVP is used to predict the motion type for the current block. The maximum displacement (Ψ) is mathematically expressed as:

$$\Psi = max(|mvp_x|, |mvp_y|) \tag{5.9}$$

and motion type is classified as:

$$MotionType = \begin{cases} Slow, & \Psi < 2 \\ Medium, & 2 \leq \Psi < 4 \\ Fast, & \Psi \geq 4 \end{cases} \tag{5.10}$$

The motion type classification for average MV distribution of all video sequences using (5.10) is illustrated in Figure 5.7. The MV distribution is already mentioned in the Table 5.1.

5.3.4 Selection of search patterns

Search patterns play a very important role in BMME [125, 126, 130]. A search pattern is responsible not only for speeding-up or checking fewer points, but also affecting the visual quality. A search pattern is selected on the basis of its shape and step-size. The shape of a search pattern should be so compact such that all possible directions are considered. A small step-size search pattern is frequently trapped in a local minimum for medium and fast motion content. It leads to inaccurate determination of MV and endures large number

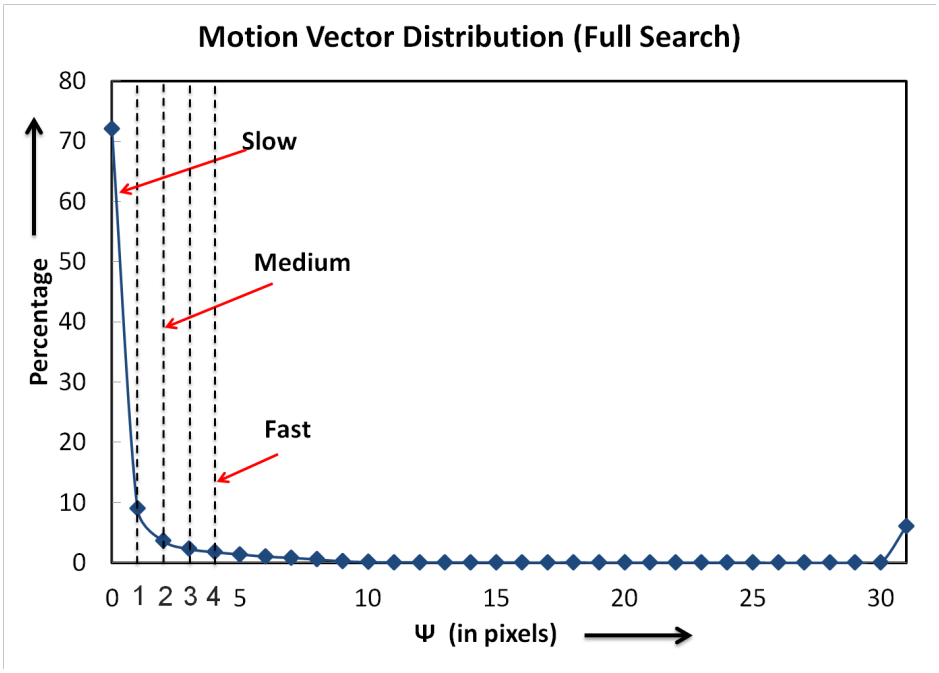


Figure 5.7: Motion vector distribution using full search (FS) with search range of ± 32

of SAD computations [116]. On the other hand, a large step-size search pattern, usually employed for slow/medium motion content, leads to excessive undesired SAD computations that increases encoding time and may even miss the global minimum [135]. Hence, a search pattern of one shape and a fixed step-size cannot handle all kinds of motion content. The proposed DAME algorithm adaptively selects the shape and step-size of different search patterns based on the motion type of video contents. This helps in evading local minimum and minimizing the number of SAD computations. Initially, the proposed DAME algorithm checks points in all directions with SDSP and CSP to avoid getting trapped in local minimum in a particular direction. After an initial omnidirectional search, it selects a directional search pattern like KSP, VHSP or HHSP to speed-up the search process and avoid unnecessary SAD computations. The search patterns employed in the proposed DAME scheme are shown in Figure 5.8. The searching process of each of the search pattern will stop at the occurrence of the early termination criteria which are defined as:

Early termination criteria:

(i) Minimum SAD value located at the centre of the pattern;

(ii) End of search window.

The selection of search patterns in the proposed DAME scheme, based on motion types, are explained below.

Slow motion

For slow motion type video content, the proposed DAME scheme uses asymmetric shaped KSP, which checks fewer points than other small search patterns such as cross, square or

diamond and yields better visual quality [187]. The search process is described below.

Step 1: If $\Psi < 2$, then motion type of the current block is slow and apply KSP and continue the search until the occurrence of early termination criteria. Otherwise, go to Section *Medium motion*.

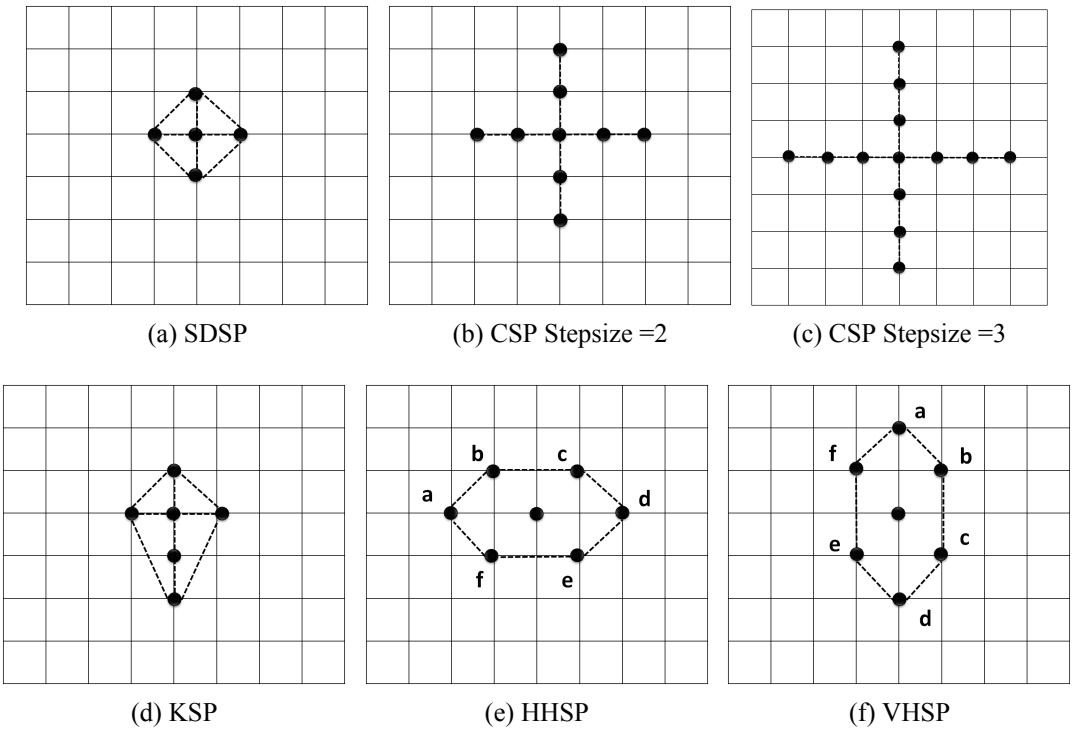
The proposed DAME search process for slow motion content is shown in the Figure 5.9(a) and the repositioning of KSP towards vertical or horizontal directions are shown in the Figure 5.10(a).

Step 2: Apply SDSP at the new search centre that is obtained from previous step. The search point which yields minimum SAD value is the best matched location for the current block.

Medium motion

For medium motion type, the proposed DAME scheme uses hybrid search patterns with larger step-size. The larger step-size improves the matching speed and also avoids getting trapped in local minimum. The DAME uses CSP and directional hexagonal search patterns (HHSP and VHSP) in this category. It is proposed to use CSP as initial search pattern to improve the search speed; and at subsequent stages, directional hexagonal search patterns are used. Hexagonal search patterns are more compact in shape and check fewer points as compared to DS [131]. The search process is described as follows:

Step 1: If $2 \leq \Psi < 4$ then motion type of the current block is medium and apply CSP (step-size $= 2$). Otherwise, go to Section *Fast motion*.



(a) SDSP (b) CSP Stepsize =2 (c) CSP Stepsize =3

(d) KSP (e) HHSP (f) VHSP

Figure 5.8: Search patterns employed in the proposed DAME scheme

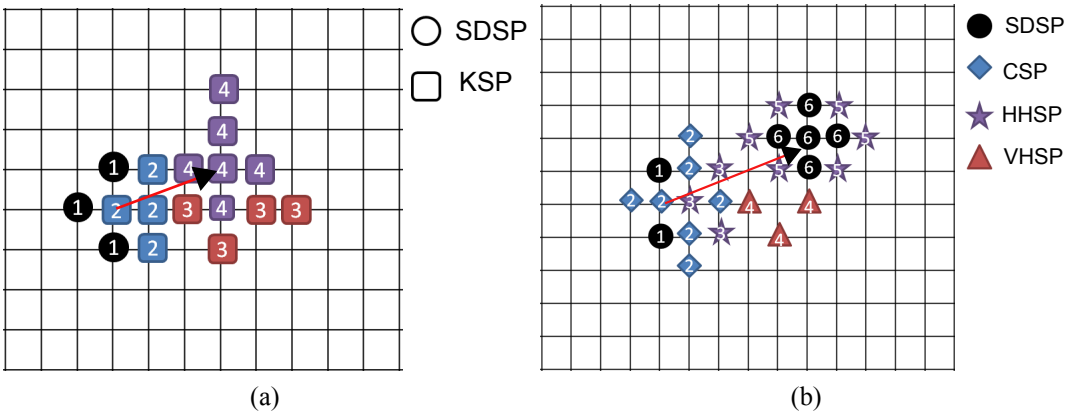


Figure 5.9: Motion vector estimation using DAME scheme: (a) MVP's $\Psi < 2$ (b) MVP's $\Psi > 2$

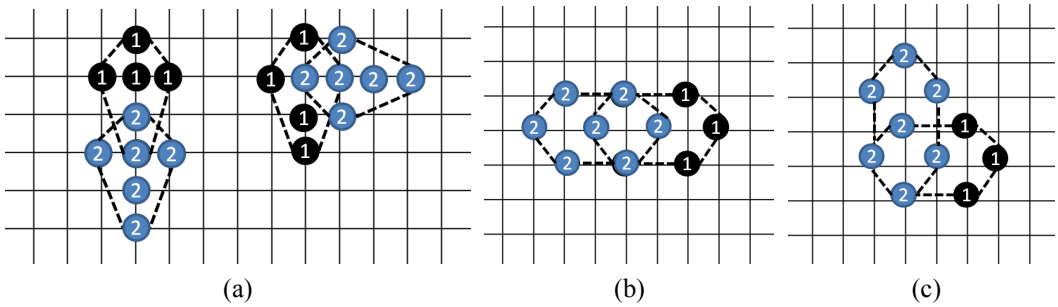


Figure 5.10: Search pattern repositioning and directional transitions: (a) KSP (b) HHSP to HHSP (c) HHSP to VHSP

Step 2: If the minimum SAD value is at the centre of CSP go to Step 4. Otherwise, go to Step 3.

Step 3: Directional hexagonal search pattern is employed to new search centre obtained from step 2. If the new centre is at horizontal axis, HHSP is used, otherwise, VHSP is considered as starting search pattern. If the minimum SAD value is located at far ends (points *a* and *d* as shown in the Figure 5.8(e) and Figure 5.8(f)) of directional hexagonal search patterns, the same pattern continues, otherwise, the pattern switches to another directional hexagonal search pattern i.e., transition from HHSP to VHSP or vice-versa.

The searching process continues until the occurrence of aforementioned early termination criteria.

The DAME search process for medium motion content is shown in the Figure 5.9(b) and the transitions of directional hexagonal search patterns towards vertical or horizontal directions (HHSP to HHSP and HHSP to VHSP) are shown in the Figure 5.10(b) and Figure 5.10(c), respectively.

Step 4: Apply SDSP at the new search centre that is obtained from previous step. The search point which yields minimum SAD value is the best matched location for the current block.

Fast motion

For fast motion type, the proposed DAME scheme applies similar procedure as used in medium motion type, already discussed in Section *Medium motion*. However, to accommodate fast motion content, the DAME algorithm uses CSP with step-size ($= 3$) as a initial search pattern.

5.4 Development of Pattern-based Modified Particle Swarm Optimization Motion Estimation (PMPSO-ME) Scheme

UES based fast BMME techniques assume that the block distortion metric such as SAD, decreases monotonically as the search moves towards global minima. These fast BMME techniques search a small subset of available set of candidate blocks and estimate a MV for the current block. However, in real-world video sequences, a large number of local minima may be present within a search window. In such a case, due to MES characteristics of distortion minimization function, these fast BMME techniques can easily be trapped to these local minima and yield poor accuracy in estimating MV. A typical example of MES characteristics is shown in Figure 5.11 for search window size of 16. To resolve the local minima problem, various population based evolutionary schemes [147, 153] are analysed to ensure the global minimum. It is found that PSO is one of most efficient techniques for BMME [150, 152].

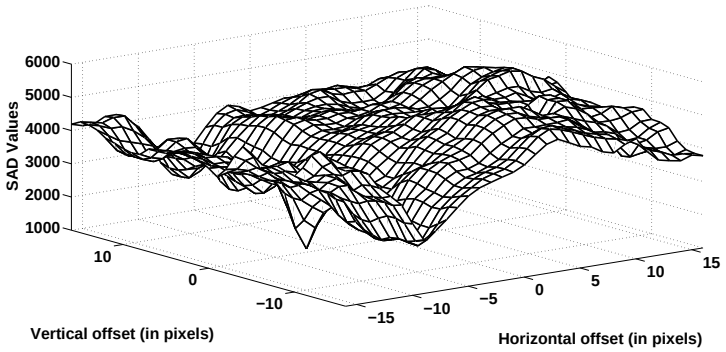


Figure 5.11: Example of Multi-modal error surface with multiple local minimum error points

5.4.1 Fundamentals of PSO based BMME

PSO is a population based, robust stochastic optimization algorithm which is inspired by the social behaviour of swarm. The scheme iteratively updates the velocities and positions of the member of swarm based on the past experience and target to achieve [144, 188].

In BMME, PSO uses the swarm intelligence to achieve the global minimum. However, since PSO is a population based optimization algorithm, accurate determination of the MV

depends on the large population of the particles, i.e., candidate block positions. Moreover, the number of iterations also plays a major role in PSO to ensure global minimum solution. In conventional PSO (CPSO), a large number of candidate search positions (also called particle)s with their initial positions and velocities are randomly chosen. Each particle has its fitness function which is SAD value for BMME. In each iteration, these particles "fly" thorough a multidimensional space of search window. The position and velocity of each particle is adaptively modified individually based on experience of its own and neighbours i.e., swarm. Each particle remembers its individual best position $pbest$ which has the best fitness function (Lowest SAD value) it has observed so far. The position of a particle which has achieved the global best fitness function in the swarm so far, is considered as global best position $gbest$. CPSO requires a large number of iterations for accomplishing a global solution [144].

Let us assume, a d-dimensional search window of size W_s and let a swarm consist of N particles $X = (\overrightarrow{\mathrm{x_1}}, \overrightarrow{\mathrm{x_2}}, \cdots, \overrightarrow{\mathrm{x_n}})$. The position of n^{th} particle is defined as:

$$\overrightarrow{\mathrm{x_n}} = \{{x_n}^1, {x_n}^2, \cdots, {x_n}^d\}' \in W_s \tag{5.11}$$

The velocity of n^{th} particle is defined as:

$$\overrightarrow{\mathrm{v_n}} = \{{v_n}^1, {v_n}^2, \cdots, {v_n}^d\}' \tag{5.12}$$

The previous best position $pbest$ of n^{th} particle is given as:

$$pbest_n = \{{p_n}^1, {p_n}^2, \cdots, {p_n}^d\}' \tag{5.13}$$

In each iteration itr, these particles "fly" and change their positions and velocities towards their $pbest$ and $gbest$. Acceleration of each moving particle, controlled by random numbers c_1 and c_2, is evaluated individually to accelerate towards individual best position $pbest$ and towards swarm's global best position $gbest$. The velocity $\overrightarrow{\mathrm{v_n}}$ and position $\overrightarrow{\mathrm{x_n}}$ of n^{th} particle for itr^{th} iteration are updated as [189]:

$$\begin{aligned} {v_n}^d(itr+1) = {v_n}^d(itr) + c_1 \times {rand_{n_1}}^d \times ({pbest_n}^d - {x_n}^d(itr)) \\ + c_2 \times {rand_{n_2}}^d \times (gbest^d - {x_n}^d(itr)) \end{aligned} \tag{5.14}$$

$${x_i}^d(itr+1) = {x_i}^d(itr) + {v_i}^d(itr) \tag{5.15}$$

where c_1 and c_2 are the positive acceleration coefficients, $rand_{n_1}$ and $rand_{n_2}$ are the random numbers which are uniformly distributed within $[0, 1]$ and itr is the number of iterations $itr = 1, 2, \cdots, itr_{max}$.

It is found that the CPSO does not have velocity control mechanism [190]. Shi and Eberhart have introduced the concept of inertia weight (I_w) [145, 189]. The parameter, I_w

restricts the influence of present velocity to next velocity for a particle in a swarm. A large value of I_w helps in global search, whereas smaller value helps in local search and hence (5.14) is modified to accommodate I_w as:

$$\begin{aligned} {v_n}^d(itr+1) = I_w \times {v_n}^d(itr) + c_1 \times {rand_{n_1}}^d \times ({pbest_n}^d - {x_n}^d(itr)) \\ + c_2 \times {rand_{n_2}}^d \times (gbest^d - {x_n}^d(itr)) \end{aligned} \tag{5.16}$$

Although it is observed that at initial stage more global exploration is advantageous and that can be achieved with higher value of I_w, but at later stages, I_w with lower value is more helpful for local search. Hence, the linear decreasing time variant inertia weight is incorporated in conventional PSO [191, 192]. The linearly decreasing time weighted PSO (LDWPSO) uses (5.16) where I_w is defined as:

$$I_w = (I_{w2} - I_{w1}) \times \left(\frac{itr_{max} - itr}{itr_{max}}\right) + I_{w2} \tag{5.17}$$

However, it is observed that the acceleration coefficients, c_1 and c_2 have primary control over velocity of a particle. c_1 is *cognitive acceleration coefficient* and c_2 is *social acceleration coefficient*. c_1 with higher value increases the movement of a particle within a search space while c_2 with higher value converges rapidly to present global best. Hence, time varying acceleration coefficients based PSO model (TVACPSO) is introduced [148]. The TVACPSO updates the velocity of a particle as:

$$\begin{aligned} {v_n}^d(itr+1) = I_w \times {v_n}^d(itr) + c_1(t) \times {r_{n_1}}^d \times ({pbest_n}^d - {x_n}^d(itr)) \\ + c_2(t) \times {r_{n_2}}^d \times (gbest^d - {x_n}^d(itr)) \end{aligned} \tag{5.18}$$

where $c_1(t)$ and $c_2(t)$ are defined as:

$$c_1(t) = (c_{1_{min}} - c_{1_{max}}) \times \frac{itr}{itr_{max}} + c_{1_{max}} \tag{5.19}$$

$$c_2(t) = (c_{2_{max}} - c_{2_{min}}) \times \frac{itr}{itr_{max}} + c_{2_{min}} \tag{5.20}$$

5.4.2 Details of PMPSO-ME

In CPSO, there is a chance that the particles may fly out of the search window and result in invalid global minimum. To restrict the particles within the search window, the velocities and positions are limited to $[v_{min}, v_{max}]^d$ and $[x_{min}, x_{max}]^d$, respectively. In general, the velocities are set to $v_{min} = x_{min}$ and $v_{max} = x_{max}$, where $x_{min} = -W_s$ and $x_{max} = W_s$ for BMME schemes. There are various schemes available in literature to resolve this fly out issue [193–195]. However, these variants of PSO either fall on modifying velocity updating equation or restricting the position within search space and hence lead to higher

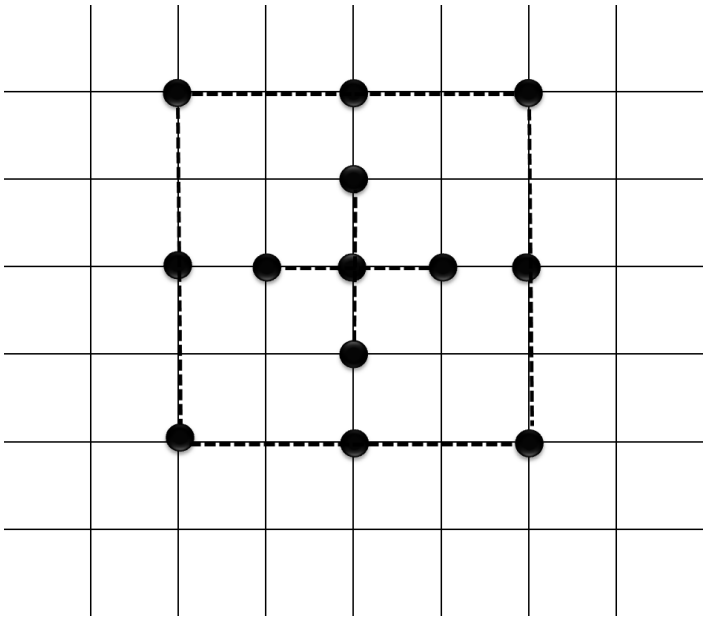


Figure 5.12: Initial particle positions in a swarm of PMPSO-ME

computational complexity.

In this chapter, we have introduced an efficient pattern-based modified PSO-ME (PMPSO-ME) scheme. The supervisor-student PSO model proposed by Liu and Qin is adopted for the proposed scheme [196].

In the proposed PMPSO-ME scheme, two fixed patterns: cross search pattern (CSP) and square search pattern (SSP) are combined to form a hybrid pattern. The locations of the hybrid pattern are initial positions of particles in swarm of PMPSO-ME scheme as shown in Figure 5.12. The swarm of particles fly in each iteration within two-dimensional search window of size W_s. The positions of the particles are best matched candidate blocks and indexed by horizontal and vertical components. We also suggest a set of early termination strategies to speed up the search process while maintaining high accuracy in estimating MV. The algorithm for implementing the proposed PMPSO-ME scheme is presented as **Algorithm 5.1**.

In PMPSO-ME, velocities of the particles are calculated by (5.14), but the positions are modulated as:

$$x_i{}^d(itr+1) = (1-\kappa) \times x_i{}^d(itr) + \kappa \times v_i{}^d(itr) \tag{5.21}$$

where κ is momentum factor $\kappa = (0,1) = \{\kappa \in \mathbb{R} \mid 0 < \kappa < 1\}$. The velocities are set to $v_{min} = x_{min}$ and $v_{max} = x_{max}$. Since the velocities are restricted within a search window and virtually considered as a position, the new position of a particle is a point in the linear equation between former position and velocity. Since the former velocity is in the range, the new position will also be in the range according to (5.21). Thus, the proposed PMPSO-ME limits the particles to fly out of search window without checking the boundary position at

Algorithm 5.1 Proposed pattern-based modified particle swarm optimization – motion estimation (PMPSO-ME) algorithm

Step 1: Initialization
Initialize the positions of all particles $X = (\overrightarrow{\mathrm{x_1}}, \overrightarrow{\mathrm{x_2}}, \cdots, \overrightarrow{\mathrm{x_n}})$ with a hybrid pattern (as shown in Figure 5.12).
Initialize the velocities of all particles $V = (\overrightarrow{\mathrm{v_1}}, \overrightarrow{\mathrm{v_2}}, \cdots, \overrightarrow{\mathrm{v_n}})$.
Evaluate the fitness (SAD) values, $\Phi = (\Phi_1, \Phi_2, \cdots, \Phi_n)$ of X.
Set X to be $pbest = (pbest_1, pbest_2, \cdots, pbest_n)$ for each particle.
Set the particle with best fitness to be $gbest$.
Set iteration $itr = 0$.
Step 2: Updating Loop
for i = 1 **to** n **do**
  Evaluate the fitness value Φ_i of a particle x_i.
  if Fitness value of $\overrightarrow{\mathrm{x_i}}$ is worse than previous iteration **then**
    Update the velocity v_i of particle x_i using (5.14).
  end if
  Update the position of particle x_i using (5.21).
  if Fitness value of x_i is better than $pbest_i$ **then**
    Set x_i to be $pbest_i$.
  end if
  if Fitness value of x_i is better than $gbest$ **then**
    Set x_i to be $gbest$.
  end if
end for
Set iteration $itr = itr + 1$.
Step 3: If termination condition is not met, GOTO Step 2, otherwise end PMPSO

every iteration.

In PMPSO-ME, the velocity (5.14) works as guiding right direction, but does not provide exact solution similar to a supervisor role. Hence, the velocity should not change at every iteration unless the direction is right. However, position described by (5.21) fine tunes itself in the given direction to determine the optimum solution. On the other hand, in other variants of PSO, the velocity is updated at every iteration. In PMPSO-ME, the velocity of each particle is updated only if the fitness of a particle in current iteration is worse than previous iteration, otherwise the velocity will be unchanged. Therefore, the proposed PMPSO-ME reduces the computational cost by obviating frequent update of velocity.

In CPSO, more number of iterations are allowed to refine the optimal solution, which lead to high computational complexity. In BMME, the number of iterations should be limited. Therefore, in the proposed PMPSO-ME, various early termination strategies are introduced to speed up the matching process. The PMPSO-ME process will stop at the occurrence of any of these early termination criteria. The early termination strategies used in this proposed PMPSO-ME scheme are as follows.

Early termination strategies:

Table 5.2: Encoder configuration in JM 18.6 reference software of H.264/AVC

Common Parameters	Inter-Coding
FrameRate = 30.0	FramesToBeEncoded = 100
DisableIntra16x16 = 1	IntraPeriod = 0
EnableIPCM = 0	IDRPeriod = 0
NumberBFrames = 0	QPISlice = 26
PicInterlace = 0	QPPSlice = {20, 26, 32, 38}
MbInterlace = 0	DisableSubpelME = 0
RDOptimization = 1	SearchRange = 32
NumberBFrames = 0	ChromaMEEnable = 0
YUVFormat = 1	PSliceSearch4x4 = 1
SourceBitDepthLuma = 8	NumberReferenceFrames = 1
SourceBitDepthChroma = 8	DisableIntraInInter = 1
Transform8x8Mode = 0	
SymbolMode = 0	

- When the total number of iterations (itr) reaches maximum value (itr_{max}).
- When SAD cost value is less than Υ_{SAD} at $gbest$.
- When $gbest$ remains same for 3 iterations successively.

5.5 Experimental Results and Discussion

With the objective of measuring the performance of the proposed DAME scheme, various experiments have been performed on H.264/AVC joint model reference software (version 18.6) [138]. For DAME algorithm, different benchmark measures are computed and compared against integral BMME schemes (FS [125], FFS [197], UMH [116], SUMH [198] and EPZS [117]) on JM reference software. Similarly, the proposed PMPSO-ME scheme is also compared with other existing PSO based BMME schemes (IPSO [150], APSO [152], PBPSO [154]) along-with FS [125] on JM reference software.

5.5.1 Experimental set-up

All experiments are carried out on standard video sequences like *Foreman, Highway,* and *Mobile*. Video sequences are categorized in terms of their resolutions as QCIF, CIF, 4CIF and HD 720p. The details of the test video sequences are listed in Table 3.5. These video sequences provide a combination of all kinds of motion contents (slow/medium/fast).

In inter-coding, all video sequences are encoded with frame pattern IPPP$\cdots$ (first I-frame and remaining all p-frames). The video sequences are encoded by a set of four quantization parameter (QP) values 20, 26, 32 and 38. The results are shown as average of four different QP values (20, 26, 32 and 38). Entropy encoding mode is set to context adaptive variable length coding (CAVLC) mode. For video visual quality, the CPSNR is used in our

Table 5.3: Bjontegaard metric[36] performance in H.264/AVC platform

	Sequence	Schemes				
		FFS [197]	UMH [116]	SUMH [198]	EPZS [117]	DAME
BD-PSNR (in dB)	Foreman	0.06	0.54	-0.76	0.83	0.85
	Highway	-0.01	-0.06	-0.10	-0.02	0.41
	Mobile	0.20	0.98	0.83	1.28	1.40
	Bus	0.27	-0.36	-1.55	1.35	0.35
	Crew	0.06	0.82	0.68	1.01	1.56
	Soccer	0.10	-0.31	-0.53	0.40	0.20
	Old town cross	0.38	-0.26	1.40	-1.61	0.49
	Park joy	1.40	1.19	1.50	2.60	1.26
	Average	**0.31**	**0.32**	**0.18**	**0.73**	**0.81**
BD-SSIM	Foreman	0.0004	0.0021	-0.0057	0.0030	0.0053
	Highway	-0.0008	-0.0012	0.0025	-0.0014	0.0078
	Mobile	0.0022	0.0101	0.0083	0.0129	0.0141
	Bus	0.0023	-0.0082	-0.0218	0.0131	0.0011
	Crew	0.0004	0.0076	0.0054	0.0077	0.0145
	Soccer	0.0011	-0.0143	-0.0083	-0.0021	-0.0033
	Old town cross	0.0070	-0.0117	0.0254	-0.0425	-0.0044
	Park joy	0.0260	0.0220	0.0259	0.0476	0.0195
	Average	**0.0048**	**0.0008**	**0.0040**	**0.0048**	**0.0068**
BD-bitrate (%)	Foreman	-0.37	-6.52	12.48	-10.37	-10.39
	Highway	0.18	1.33	14.00	-0.45	-4.38
	Mobile	-3.83	-15.28	-13.11	-19.24	-21.05
	Bus	-4.74	6.50	29.26	-20.38	-6.14
	Crew	-0.43	-14.69	-12.47	-17.97	-25.31
	Soccer	-0.33	-3.92	13.22	-8.81	-3.17
	Old town cross	-0.89	-8.32	7.82	-11.29	-11.50
	Park joy	-19.78	-17.97	-22.24	-35.50	-19.93
	Average	**-3.77**	**-7.36**	**3.62**	**-15.50**	**-12.73**

experiments. The detailed encoder configuration for JM18.6 is listed on Table 5.2. In our experiment, value of Υ_{SAD} is set to 256.

5.5.2 Experimental results of DAME algorithm

Experiment 1: Bjontegaard metrics performance

In this experiment, other existing BMME schemes are compared with the proposed DAME scheme with respect to BD-PSNR, BD-SSIM and BD-bitrate with respect to FS. The comparative analysis is tabulated in Table 5.3. In Bjontegaard metric, positive numbers in BD-PSNR and BD-SSIM represent gain, while negative numbers in BD-bitrate show reduction in bit-rate. The R-D curves of *Foreman, Mobile, Crew* and *Old Town Cross* sequences, shown in Figures 5.13, 5.14, 5.15 and 5.16, respectively, exhibit performance comparisons between state-of-the-art techniques and the proposed algorithm.

It may be observed form Table 5.3 that DAME outperforms other BMME techniques with

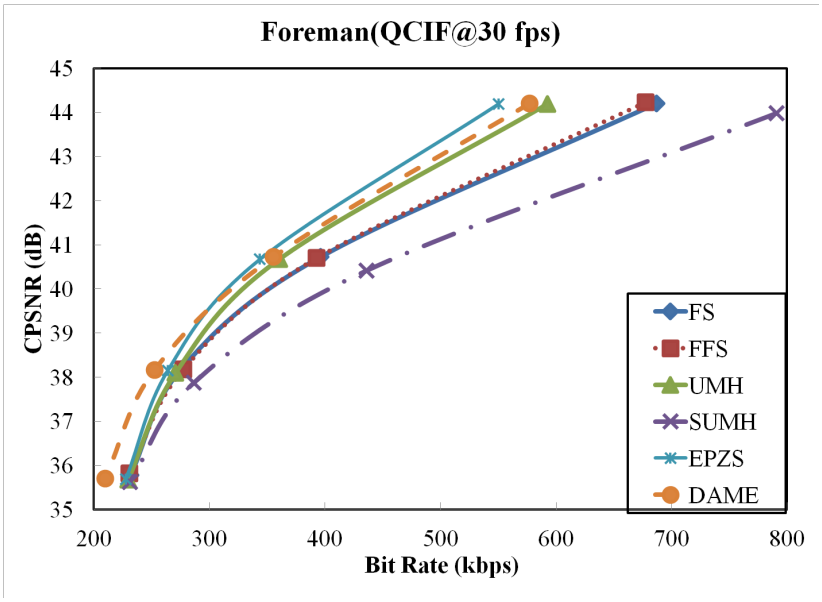


Figure 5.13: Rate-distortion curves for *Foreman* sequence

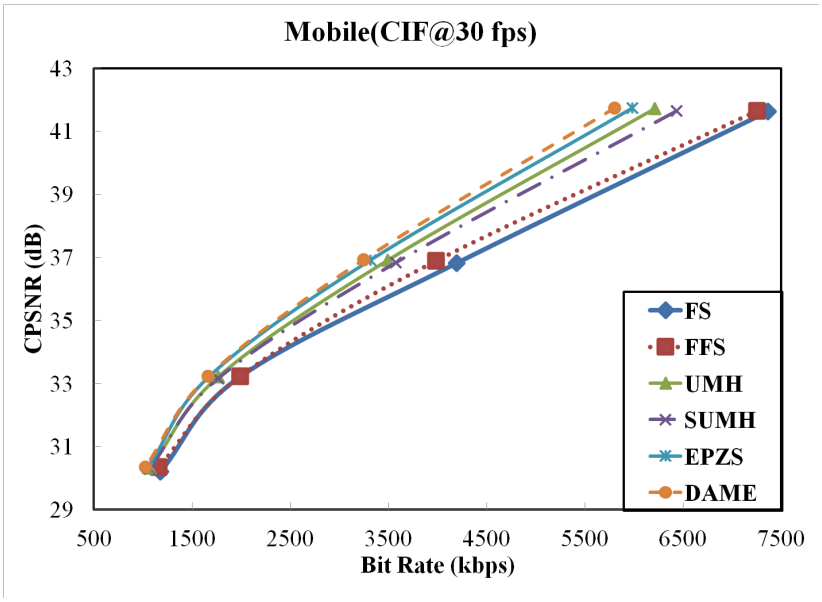


Figure 5.14: Rate-distortion curves for *Mobile* sequence

respect to BD-PSNR (or equivalently BD-bitrate). The proposed DAME algorithm achieves improvement in BD-PSNR of 0.81 dB (or equivalently 12.73% reduction in BD-bitrate) on average. It is also noticed that DAME achieves highest BD-PSNR gain of 1.56 dB (or equivalently 25.31% reduction in BD-bitrate) for *Crew* sequence.

Experiment 2: Performance analysis of number of search points

Table 5.4 shows the comparison between the proposed DAME with respect to number of search points and other existing fast BMME schemes. It is observed that the proposed DAME scheme outperforms other schemes. UMH (666.39%), SUMH (117.90%) and EPZS

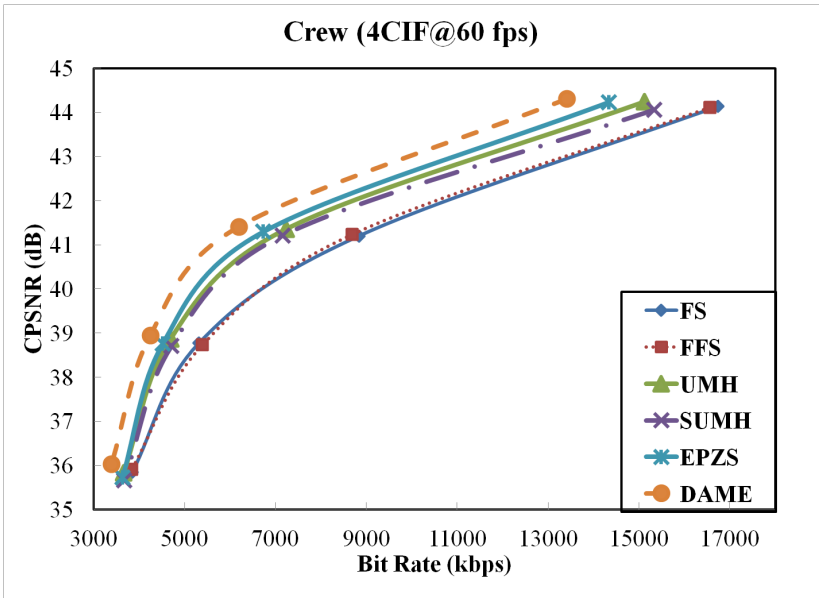


Figure 5.15: Rate-distortion curves for *Crew* sequence

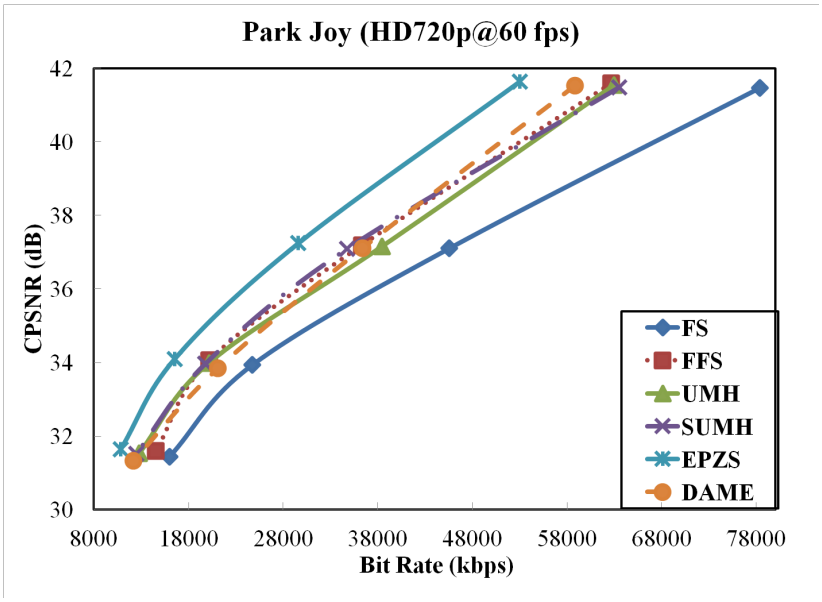
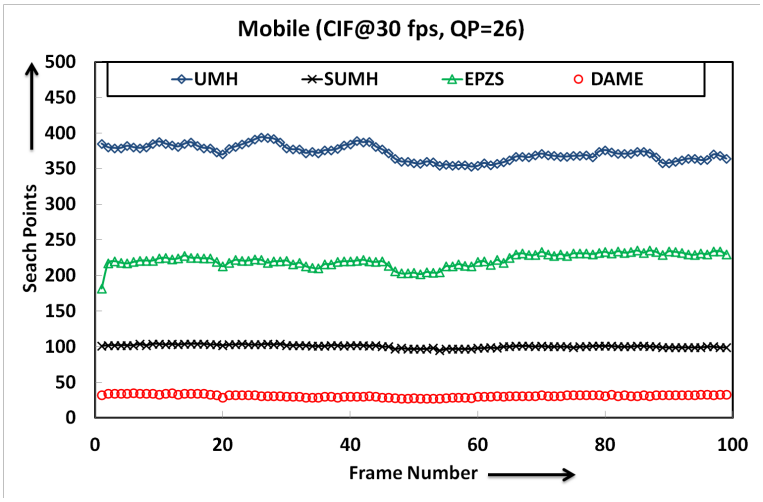


Figure 5.16: Rate-distortion curves for *Old Town Cross* sequence

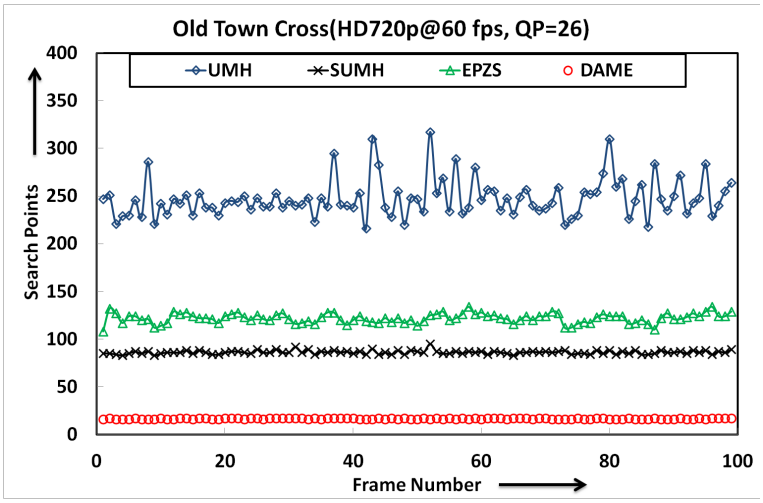
(271.19%) perform more number of search points than the proposed DAME scheme. The comparison of average number of search points per macroblock per frame is also shown in Figure 5.17 for *Mobile* and *Old Town Cross* sequences.

Experiment 3: Analysis of threshold (Υ_{SAD}) values

As mentioned earlier, Υ_{SAD} value is set to 256 in our experiment. Table 5.5 shows a detailed performance comparison for different threshold values at QP equals to 26. It can be observed that the proposed DAME scheme with Υ_{SAD} equal to 256 yields best performance than for any other threshold values. Lowering threshold value below 256 leads to significant



(a)



(b)

Figure 5.17: Comparison of average number of search points per macroblock per frame for *Mobile* and *Old Town Cross* sequences at QP=26

increment in number of SAD computations whereas increasing the threshold results lesser SAD computation. However, in both the cases, output bit-rates are increased considerably.

Experiment 4: MV distribution

To study the characteristics of MV distribution, the comparative analysis of average MV distribution of all video sequences for FS and the proposed DAME is given in Figure 5.18. Also, Table 5.6 presents MV distribution achieved by the proposed DAME scheme which is almost similar to MV distribution of FS. One reason for the small difference in MV distribution against FS is the early termination techniques that reduce number of SAD computations considerably.

Table 5.4: Performance comparison in terms of number of search points per macroblock

	Sequence	Schemes					
		FS [125]	FFS [197]	UMH [116]	SUMH [198]	EPZS [117]	DAME
Search points	Foreman	67600.00	67600.00	358.25	90.50	165.00	25.25
	Highway	67600.00	67600.00	323.25	82.00	104.50	17.25
	Mobile	67600.00	67600.00	363.50	103.50	212.75	36.50
	Bus	67600.00	67600.00	396.75	119.50	205.00	81.75
	Crew	67600.00	67600.00	343.00	96.75	165.00	24.00
	Soccer	67600.00	67600.00	353.50	96.00	159.00	30.75
	Old Town Cross	67600.00	67600.00	278.25	86.75	120.75	26.75
	Park Joy	67600.00	67600.00	377.00	118.50	221.00	122.25
	Average	**67600.00**	**67600.00**	**349.19**	**99.19**	**169.13**	**45.56**
ΔSearch points (in %)	Foreman	267622.77	267622.77	1318.81	258.42	553.47	0.00
	Highway	391784.06	391784.06	1773.91	375.36	505.80	0.00
	Mobile	185105.48	185105.48	895.89	183.56	482.88	0.00
	Bus	82591.13	82591.13	385.32	46.18	150.76	0.00
	Crew	281566.67	281566.67	1329.17	303.13	587.50	0.00
	Soccer	219737.40	219737.40	1049.59	212.20	417.07	0.00
	Old Town Cross	252610.28	252610.28	940.19	224.30	351.40	0.00
	Park Joy	55196.52	55196.52	208.38	-3.07	80.78	0.00
	Average	**148267.63**	**148267.63**	**666.39**	**117.70**	**271.19**	**0.00**

Table 5.5: Performance comparison of DAME scheme for different threshold (Υ_{SAD}) values at QP = 26

	Threshold	**Sequence**								
		Foreman	Highway	Mobile	Bus	Crew	Soccer	Old Town Cross	Park Joy	Average
PSNR-Y	128	38.16	39.17	36.37	36.81	38.91	38.06	37.28	36.73	**37.69**
	256	38.10	39.16	36.36	36.77	38.93	38.05	37.29	36.72	**37.67**
	384	38.08	39.14	36.37	36.78	38.93	38.05	37.28	36.73	**37.67**
Bitrate	128	372.06	313.99	3434.74	3259.34	6529.89	6586.52	9917.93	46116.59	**9566.38**
	256	355.65	303.15	3247.75	2972.98	6197.95	6004.77	9304.77	36365.70	**8094.09**
	384	379.27	323.15	3426.48	3444.10	6484.77	6868.78	9787.03	47912.84	**9828.30**
Search points	128	36.00	21.00	49.00	112.00	35.00	46.00	25.00	162.00	**60.75**
	256	21.00	17.00	31.00	77.00	22.00	25.00	16.00	125.00	**41.75**
	384	17.00	16.00	23.00	69.00	18.00	20.00	16.00	104.00	**35.38**

Experiment 5: Analysis of encoding time complexity

The proposed DAME scheme performs less number of SAD computations and results in reduction in encoding time. Table 5.7 shows the comparative encoding time analysis of DAME with different BMME scheme for various video sequences with respect to FS. It is observed that the proposed DAME outperforms other existing BMME schemes. The average ΔT of DAME is -0.42, i.e., the proposed DAME yields encoder time ratio of 58% with respect to FS encoding time, whereas the average ΔT equals to 0.72, -0.39 , -0.34 and -0.40 for FFS, UMH, SUMH and EPZS, respectively. Therefore, it may be concluded that the proposed DAME scheme requires less encoding time and less number of SAD computation without degrading visual quality.

Table 5.6: MV distribution based on maximum displacement categories using proposed DAME scheme for search range of ± 32

Ψ (in Pixels)	0	1	2 or 3	4 or more
Foreman	89.70	6.44	3.03	0.80
Highway	94.21	3.62	2.13	0.03
Mobile	80.07	16.19	2.91	0.82
Bus	74.60	5.22	7.39	12.11
Crew	90.21	3.70	2.76	3.20
Soccer	84.42	8.08	4.33	3.18
Park Joy	68.93	2.51	2.67	26.69
Old Town Cross	84.28	5.55	3.76	6.41
Average	**83.30**	**6.41**	**3.62**	**6.65**

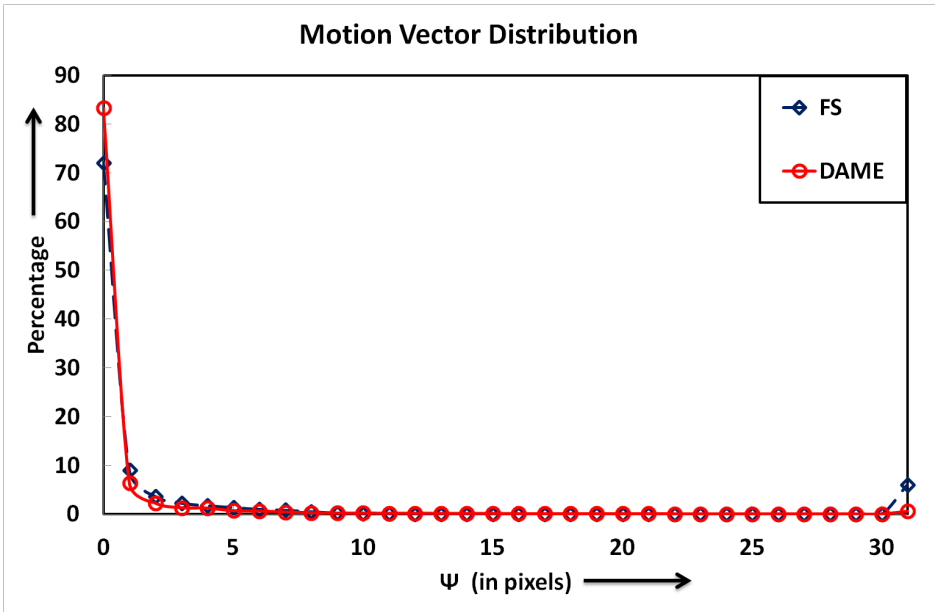


Figure 5.18: Comparison of overall motion vector distribution of Full search and the proposed DAME scheme

To analyse the efficacy of the proposed scheme thoroughly, comparison of ME time (T_{me}) is also given in Table 5.8. It can be observed that the proposed DAME scheme significantly reduces the ME time as compared to FS and FFS schemes and also outperforms EPZS scheme. It can be observed that SUMH scheme outperforms the proposed DAME scheme and yields 1.98 second average ME time as compared to 3.79 second average ME time required by the proposed DAME scheme. However, it can be observed from the other experimental results that the SUMH scheme has compromised with the PSNR, SSIM and RD-curves for the improvement in ME time performance. In other words, this scheme is fast but not an accurate motion estimation algorithm.

Table 5.7: Performance comparison in terms of encoding time

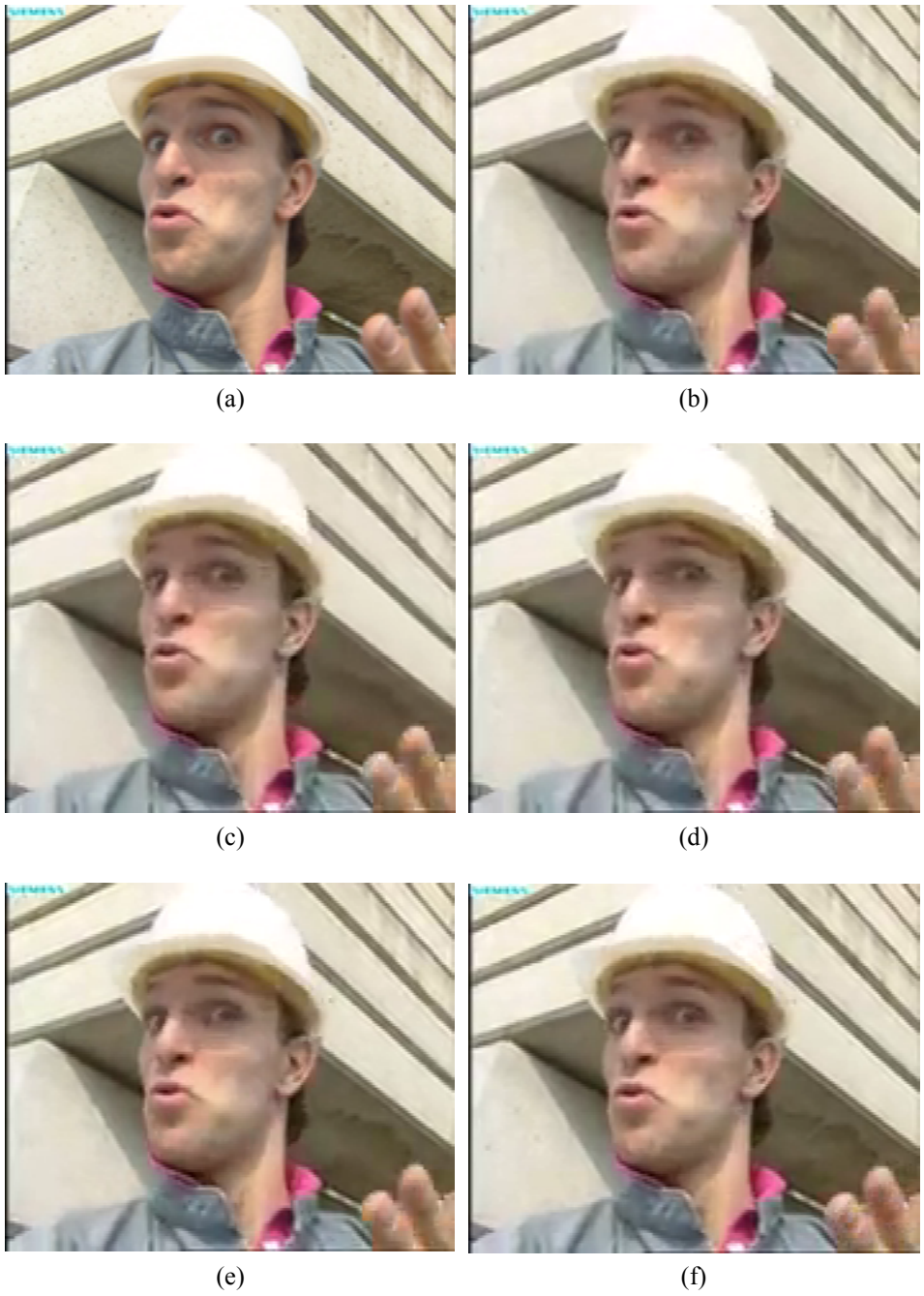
	Sequence	Schemes					
		FS [125]	FFS [197]	UMH [116]	SUMH [198]	EPZS [117]	DAME
T (in Sec.)	Foreman	8.86	16.07	5.51	6.28	5.50	5.17
	Highway	7.90	17.69	5.44	5.84	5.48	5.34
	Mobile	65.08	88.77	38.59	39.48	38.17	35.05
	Bus	52.51	79.65	33.83	38.73	29.84	33.20
	Crew	166.47	299.47	99.58	103.22	100.06	93.33
	Soccer	153.49	292.55	94.45	103.75	93.89	92.97
	Old Town Cross	354.89	673.61	204.95	229.16	206.77	189.96
	Park Joy	655.67	818.39	373.27	373.11	**340.18**	351.70
	Average	**183.11**	**285.78**	**106.95**	**112.44**	**102.49**	**100.84**
ΔT	Foreman	0.00	0.81	-0.38	-0.29	-0.38	-0.42
	Highway	0.00	1.24	-0.31	-0.26	-0.31	-0.32
	Mobile	0.00	0.36	-0.41	-0.39	-0.41	-0.46
	Bus	0.00	0.52	-0.36	-0.26	-0.43	-0.37
	Crew	0.00	0.80	-0.40	-0.38	-0.40	-0.44
	Soccer	0.00	0.91	-0.38	-0.32	-0.39	-0.39
	Old Town Cross	0.00	0.90	-0.42	-0.35	-0.42	-0.46
	Park Joy	0.00	0.25	-0.43	-0.43	-0.48	-0.46
	Average	**0.00**	**0.72**	**-0.39**	**-0.34**	**-0.40**	**-0.42**

Table 5.8: Performance comparison in terms of motion estimation time T_{me}

	Sequence	FS [125]	FFS [197]	UMH [116]	f SUMH [198]	EPZS [117]	DAME
T_{me} **(in Sec.)**	Foreman	3.59	12.55	0.27	0.11	0.35	0.25
	Highway	2.67	12.51	0.22	0.07	0.26	0.19
	Mobile	25.21	50.41	1.19	0.58	1.52	1.12
	Bus	22.10	49.89	1.08	0.65	1.54	1.12
	Crew	67.77	203.28	4.03	2.00	5.40	3.93
	Soccer	60.98	204.05	3.87	2.01	5.46	4.07
	Old Town Cross	150.47	466.75	7.89	4.30	10.91	8.66
	Park Joy	268.31	470.55	10.43	6.13	14.03	10.99
	Average	**75.14**	**183.75**	**3.62**	**1.98**	**4.93**	**3.79**
ΔT_{me}	Foreman	0.00	-2.50	0.93	0.97	0.90	0.93
	Highway	0.00	-3.68	0.92	0.98	0.90	0.93
	Mobile	0.00	-1.00	0.95	0.98	0.94	0.96
	Bus	0.00	-1.26	0.95	0.97	0.93	0.95
	Crew	0.00	-2.00	0.94	0.97	0.92	0.94
	Soccer	0.00	-2.35	0.94	0.97	0.91	0.93
	Old Town Cross	0.00	-2.10	0.95	0.97	0.93	0.94
	Park Joy	0.00	-0.75	0.96	0.98	0.95	0.96
	Average	**0.00**	**-1.95**	**0.94**	**0.97**	**0.92**	**0.94**

Experiment 6: Subjective performance

The subjective performance comparison of DAME scheme against existing competitive schemes is shown in Figure 5.19 for *Foreman* sequence. It is clearly seen that the finger tips, eyes and collar boundaries are visually more prominent as compared to those reconstructed



(a) (b)

(c) (d)

(e) (f)

Figure 5.19: Subjective performance reconstructed frame using DAME and other existing competitive schemes in *Foreman* sequence: a) Original frame, b) FS (144.14 kbps, 38.23 dB), c) HEX (134.07 kbps, 37.62 dB), d) SHEX (130.44 kbps, 37.63 dB), e) EPZS (134.14 kbps, 38.15 dB) and f) DAME (133.18 kbps, 38.53 dB)

by its competitive schemes. Only EPZS exhibits somewhat similar visual performance.

5.5.3 Experimental results of PMPSO-ME

To determine the efficiency of the proposed PMPSO-ME scheme, different experiments are conducted and outputs are compared with the other existing schemes. The total number of iterations (itr_{max}) is set to 20. In our experiment, value of Υ_{SAD} is set to 256; I_w equals

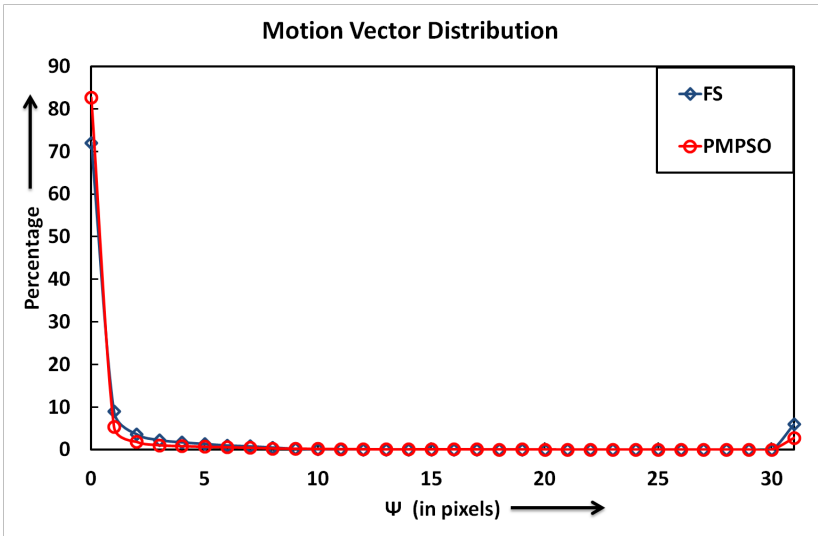


Figure 5.20: Comparison of overall motion vector distribution of Full search and the proposed PMPSO-ME scheme

to 0.5; $c_{1_{min}}, c_{1_{max}}, c_{2_{min}}$ and $c_{2_{max}}$ are set to 0.5, 2.5, 0.5 and 2.5, respectively. The MSE threshold value Υ_{MSE} for PBPSO is set to 7. It can be observed from Figure 5.20 that the MV distribution for PMPSO-ME scheme is almost similar to that for FS on average for all video sequences.

Experiment 1: Bjontegaard metrics performance

The BD-PSNR and BD-SSIM (or equivalent BD-bitrate %), with respect to FS, is computed to determine the compression performance of the proposed PMPSO-ME. The BD-PSNR (or equivalent BD-bitrate %) is summarized for all video sequences (as mentioned in Table 3.5) in Table 5.9. Further, R-D curves for the video sequence *Foreman, Mobile, Crew* and *Old Town Cross* are shown in Figures 5.21, 5.22, 5.23 and 5.24, respectively. It can be observed that the proposed PMPSO-ME outperformed other schemes and yields a BD-PSNR of 0.49 dB (or equivalently BD-bitrate -7.98%). However, other comparative schemes show loss in BD-PSNR upto -0.09 dB (or equivalently BD-bitrate -3.50%). The R-D curves exhibit that the gap between the proposed PMPSO is more at higher bit-rate than lower bit-rates. For *Crew* sequence, PMPSO achieves significant improvement of 1.06 dB gain in BD-PSNR (or equivalently BD-bitrate -18.60%).

Experiment 2: Performance analysis of number of search points

The number of search points to find the best matched block is a very important parameter in BMME. If the number of search points increases, then it leads to more SAD computations which consequently reduces the encoding time. The comparative analysis of number of search points per macroblock is tabulated in Table 5.10. As discussed earlier, apart from total number of iterations, the proposed early termination techniques are also employed on

Table 5.9: Bjontegaard metric[36] performance in H.264/AVC platform

	Sequence	Schemes			
		IPSO [150]	APSO [152]	PBPBO [154]	PMPSO
BD-PSNR (in dB)	Foreman	0.30	0.32	0.34	0.34
	Highway	-0.04	-0.01	-0.03	0.18
	Mobile	0.75	0.89	0.78	1.07
	Bus	0.18	0.20	0.11	0.32
	Crew	0.42	0.44	0.44	1.06
	Soccer	-0.02	-0.08	-0.11	0.21
	Old town cross	-1.43	-1.82	-1.61	0.36
	Park joy	-0.53	-0.63	-0.18	0.34
	Average	**-0.05**	**-0.09**	**-0.03**	**0.49**
BD-SSIM	Foreman	0.0009	0.0011	0.0015	0.0011
	Highway	0.0008	0.0010	0.0010	0.0045
	Mobile	0.0075	0.0088	0.0079	0.0111
	Bus	-0.0003	-0.0007	-0.0010	0.0006
	Crew	0.0038	0.0039	0.0037	0.0105
	Soccer	-0.0038	-0.0043	-0.0046	0.0017
	Old town cross	-0.0324	-0.0394	-0.0353	-0.0058
	Park joy	-0.0269	-0.0314	-0.0211	-0.0059
	Average	**-0.0063**	**-0.0076**	**-0.0060**	**0.0022**
BD-bitrate (%)	Foreman	-3.07	-3.34	-3.67	-3.73
	Highway	3.92	3.66	3.62	1.23
	Mobile	-11.89	-14.01	-12.61	-16.51
	Bus	-3.48	-4.04	-2.47	-5.86
	Crew	-7.55	-8.65	-8.87	-18.60
	Soccer	2.63	3.78	4.11	1.72
	Old town cross	-8.20	-7.63	-7.85	-8.42
	Park joy	-5.53	2.21	-7.19	-13.67
	Average	**-4.15**	**-3.50**	**-4.36**	**-7.98**

comparative schemes. It can be noticed that the proposed PMPSO outperforms other existing schemes. For instance, IPSO [150] checks 9.30% more search points than the proposed PMPSO. Similarly, APSO [152] and PBPSO [154]) match 12.25% and 11.40% more number of search points to determine the best matched block. The comparisons of average number of search points for *Mobile* and *Old Town Cross* video sequences for all frames at QP equals to 26 are shown in Figure 5.25. It is found that the number of search points overshoot for multiple frames of *Mobile* sequence for other existing schemes, whereas the proposed PMPSO checks less number of search points for all frames. Similarly, for *Old Town Cross* sequence, the number of search point is considerably less for proposed PMPSO than other existing schemes.

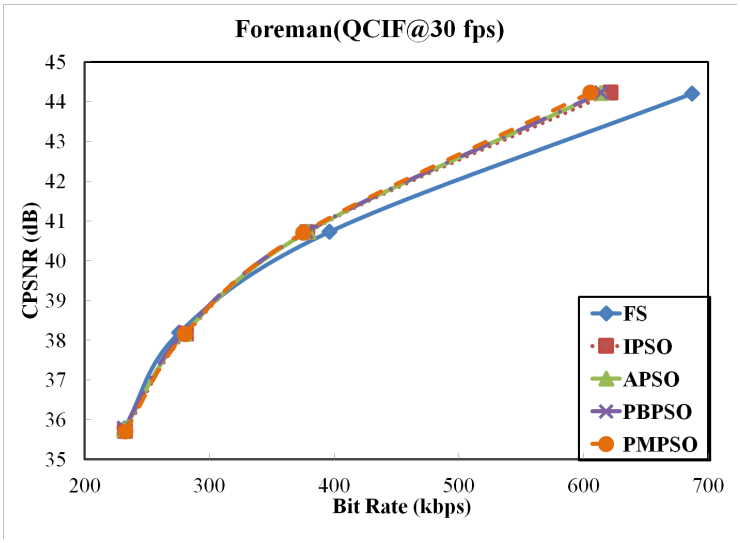


Figure 5.21: Rate-distortion curves for *Foreman* sequence

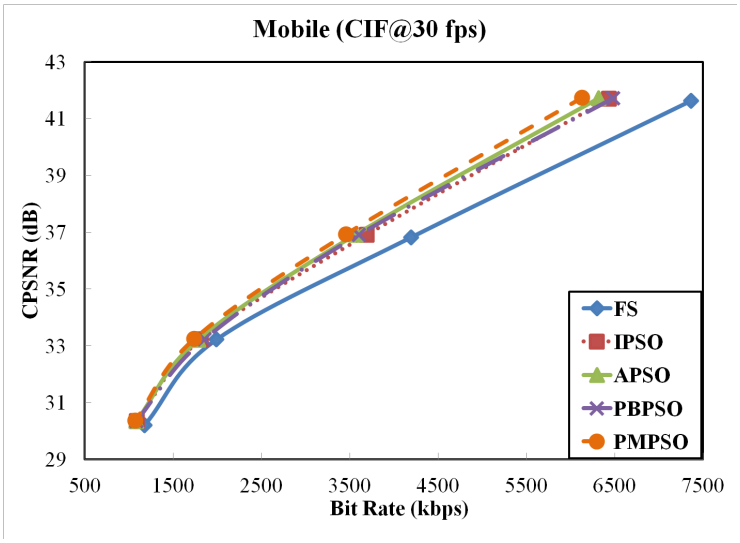


Figure 5.22: Rate-distortion curves for *Mobile* sequence

Experiment 3: Analysis of threshold (Υ_{SAD}) values

For the proposed PMPSO-ME scheme, Υ_{SAD} value is set to 256. A comparative analysis is performed for different threshold values and summarized in Table 5.11. It is observed that the proposed PMPSO-ME scheme outperforms with Υ_{SAD} equals to 256 than other threshold values. It can be seen that by increasing the threshold value, the number of search points reduces. But, unfortunately, the bit-rate increases which degrades the compression performance. If the threshold value is reduced below 256, the number of search points is increased considerably and bit-rate also increases which reduces the compression efficiency.

Experiment 4: Analysis of encoding time complexity

The proposed PMPSO-ME uses various early termination techniques and leads to check less number of search points for the best matched block. The less computational complexity

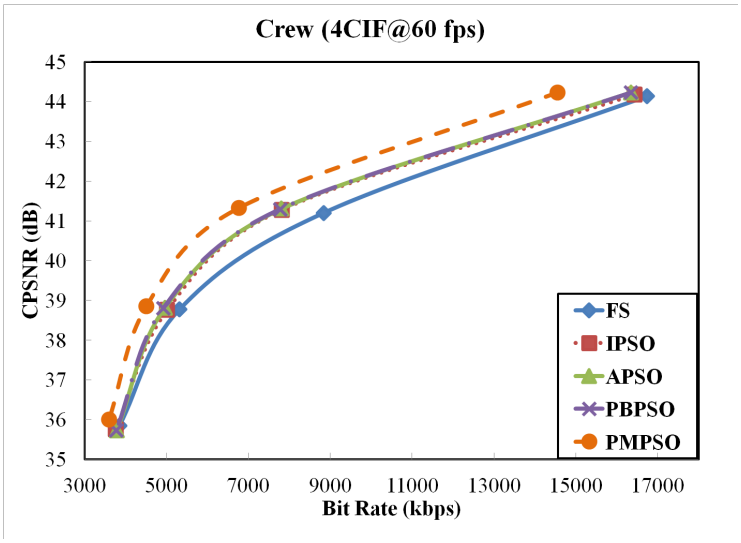


Figure 5.23: Rate-distortion curves for *Crew* sequence

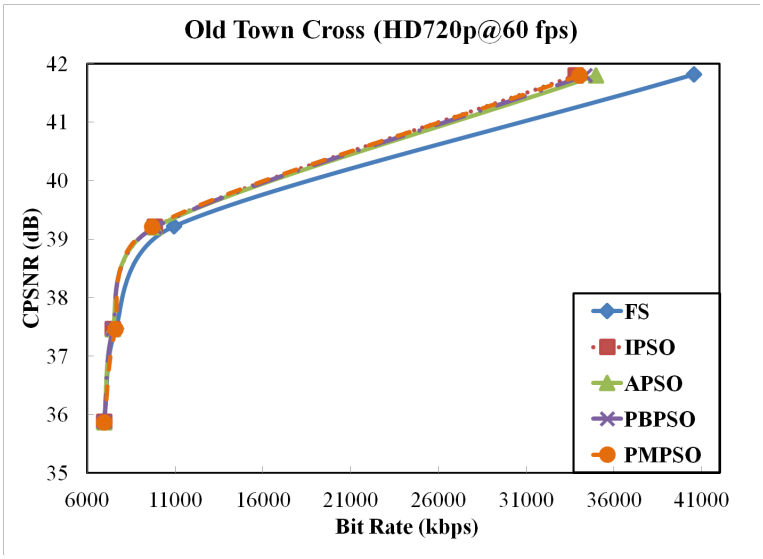
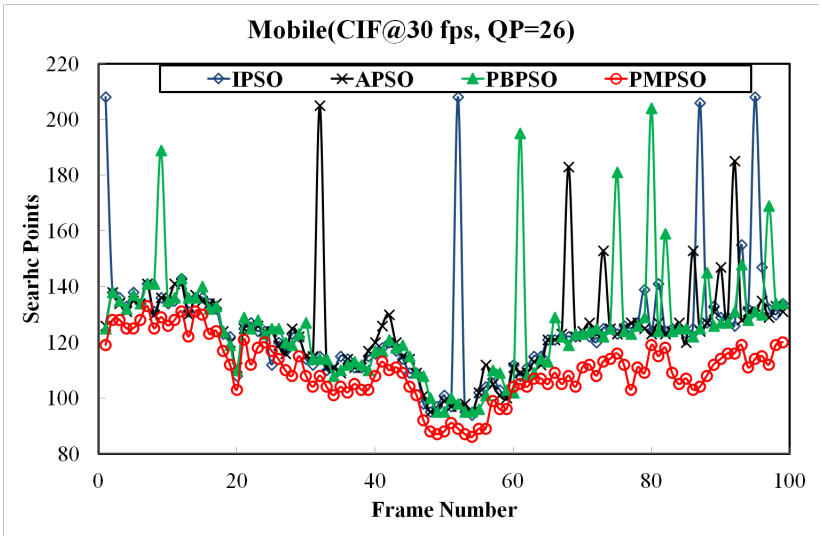


Figure 5.24: Rate-distortion curves for *Old Town Cross* sequence

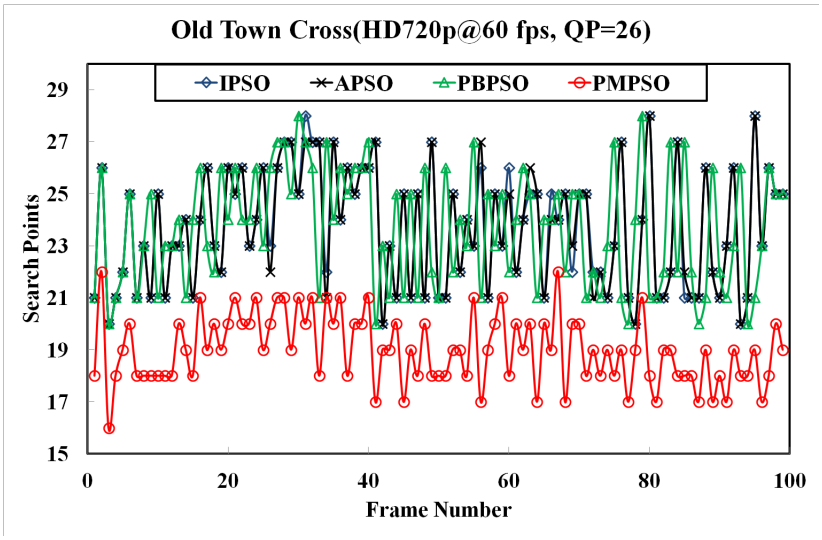
results in reduced encoding time. The comparative performance analysis of the proposed PMPSO-ME with other existing schemes is presented in Table 5.12 and Table 5.13. It can be observed that the proposed PMPSO-ME takes considerably less encoding time than other schemes. The average ΔT for PMPSO-ME is equal to -0.40 whereas it is $-0.33, -0.33$ and -0.34 for IPSO, APSO and PBPSO, respectively. Hence, the proposed PMPSO-ME encodes a video at 40% faster rate compared to the FS scheme.

Experiment 5: Subjective performance

The subjective performance of PMPSO-ME and other existing competitive schemes is shown in Figure 5.26 for *Foreman* sequence. It is clearly seen that the proposd PMPSO-ME yields better visual performance and hence, outperforms other schemes.



(a)



(b)

Figure 5.25: Comparison of average number of search points per macroblock per frame for *Mobile* and *Old Town Cross* sequences at QP=26.

5.6 Conclusion

In this chapter, we have proposed two efficient BMME schemes to reduce temporal redundancy among video frames. One is UES based DAME and another, PMPSO-ME based on MES. The DAME uses hybrid search patterns that comprises SDSP, KSP, CSP, HHSP and VHSP. It proposes to select different search pattern based on motion content which is predicted by MVP. The proposed DAME scheme significantly reduces the search points and minimizes the computational cost while maintaining visual quality. Moreover, it yields superior performance in comparison to other state-of-the-art BMME approaches. The PMPSO-ME is an evolutionary method based on PSO. Various early termination techniques are applied to achieve lower computational complexity without degrading the visual quality.

Table 5.10: Performance comparison in terms of number of search points per macroblock

	Sequence	Schemes				
		FS [125]	IPSO [150]	APSO [152]	PBPSO [154]	**PMPSO**
Search points	Foreman	67600.00	55.00	53.50	54.25	49.25
	Highway	67600.00	23.50	27.50	27.25	24.00
	Mobile	67600.00	148.00	145.25	148.25	137.25
	Bus	67600.00	152.50	154.00	153.75	140.50
	Crew	67600.00	68.00	68.50	66.00	56.00
	Soccer	67600.00	46.75	47.75	48.00	43.25
	Old Town Cross	67600.00	35.50	37.00	36.00	34.50
	Park Joy	67600.00	368.00	378.75	370.75	317.25
	Average	**67600.00**	**112.16**	**114.03**	**113.03**	**100.25**
ΔSearch points (%)	Foreman	137158.88	11.68	8.63	10.15	0.00
	Highway	281566.67	-2.08	14.58	13.54	0.00
	Mobile	49153.19	7.83	5.83	8.01	0.00
	Bus	48013.88	8.54	9.61	9.43	0.00
	Crew	120614.29	21.43	22.32	17.86	0.00
	Soccer	156200.58	8.09	10.40	10.98	0.00
	Old Town Cross	195842.03	2.90	7.25	4.35	0.00
	Park Joy	21208.12	16.00	19.39	16.86	0.00
	Average	**126219.70**	**9.30**	**12.25**	**11.40**	**0.00**

Table 5.11: Performance comparison of PMPSO-ME for different threshold (Υ_{SAD}) values at QP = 26

	Threshold	**Sequence**								
		Foreman	**Highway**	**Mobile**	**Bus**	**Crew**	**Soccer**	**Old Town Cross**	**Park Joy**	**Average**
PSNR	128	38.07	39.19	36.38	36.8	38.9	38.08	37.29	36.72	**37.68**
	256	38.11	39.16	36.38	36.79	38.90	38.07	37.29	36.71	**37.68**
	384	38.08	39.16	36.37	36.75	38.90	38.04	37.28	36.71	**37.66**
Bitrate	128	380.82	370.52	3479.68	3685.79	7432	6294.18	15888.1	38466.04	**9499.64**
	256	373.43	363.85	3101.17	3172.62	7352.21	6669.85	14566.44	34388.69	**8748.53**
	384	456.98	397.26	3826.07	4503.59	7455.10	6997.33	15787.30	39443.69	**9858.42**
Search points	128	75.00	37.00	203.00	185.00	62.00	61.00	55.00	275.00	**119.13**
	256	35.00	22.00	115.00	120.00	56.00	29.00	20.00	317.00	**89.25**
	384	21.00	18.00	60.00	84.00	27.00	21.00	16.00	173.00	**52.50**

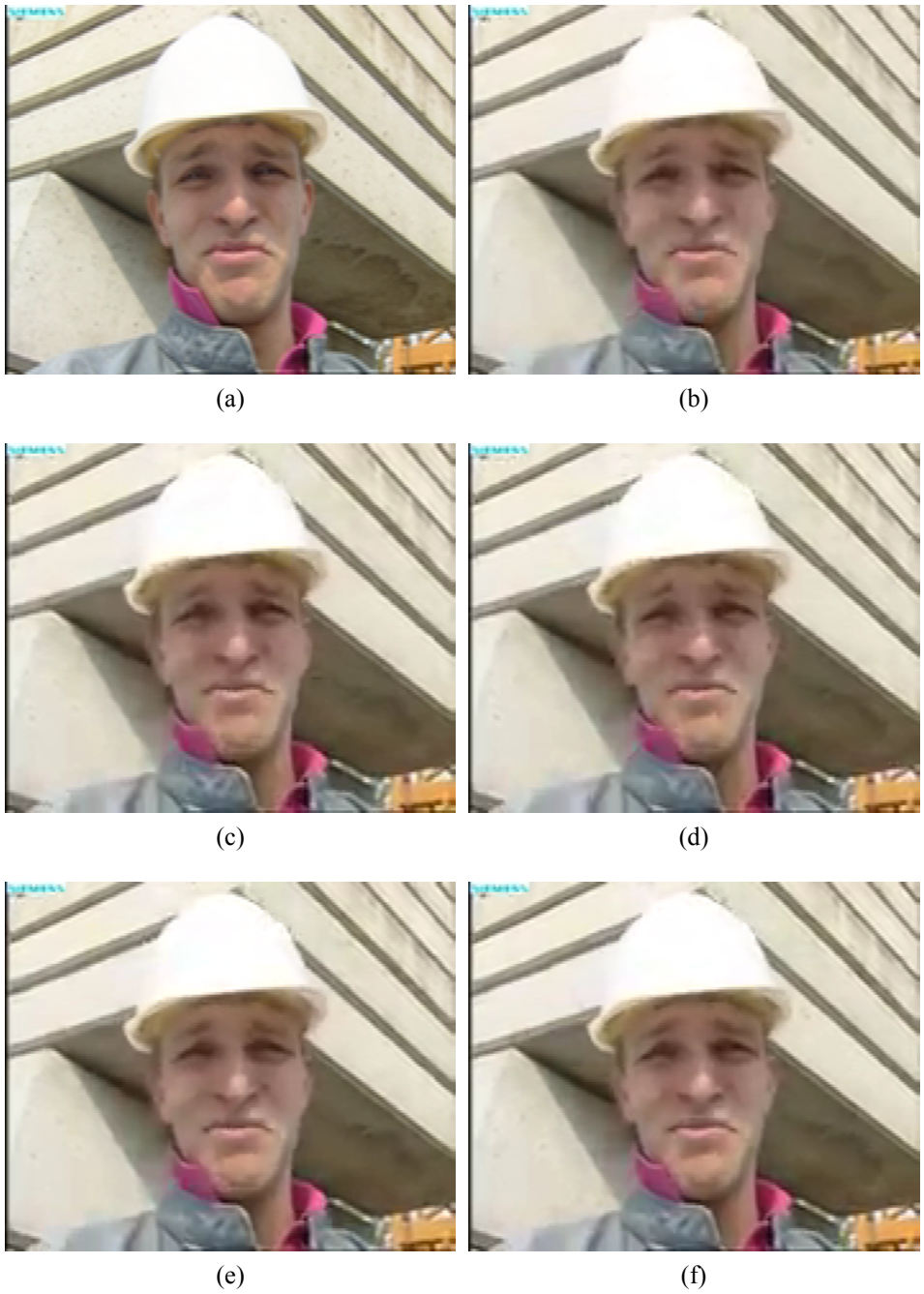
The proposed DAME or PMPSO-ME and efficient transform schemes are expected to lead to further improvement in compression performance and reduction in encoding time in rich video content based applications like HDTV broadcasting, security and surveillance.

Table 5.12: Performance comparison in terms of encoding time

	Sequence	Schemes				
		FS [125]	**IPSO** [150]	**APSO** [152]	**PBPSO** [154]	**PMPSO**
T (in Sec.)	Foreman	8.86	5.61	5.56	5.54	5.11
	Highway	7.90	5.51	5.47	5.45	5.00
	Mobile	65.08	39.51	38.98	39.29	33.93
	Bus	52.51	36.56	36.39	36.40	31.71
	Crew	166.47	103.91	103.16	102.52	93.72
	Soccer	153.49	97.18	96.85	96.71	92.78
	Old Town Cross	354.89	212.81	213.50	211.88	201.55
	Park Joy	655.67	559.98	562.25	558.60	498.69
	Average	**183.11**	**132.63**	**132.77**	**132.05**	**121.56**
ΔT	Foreman	0.00	-0.37	-0.37	-0.37	-0.42
	Highway	0.00	-0.30	-0.31	-0.31	-0.37
	Mobile	0.00	-0.39	-0.40	-0.40	-0.48
	Bus	0.00	-0.30	-0.31	-0.31	-0.40
	Crew	0.00	-0.38	-0.38	-0.38	-0.44
	Soccer	0.00	-0.37	-0.37	-0.37	-0.40
	Old Town Cross	0.00	-0.40	-0.40	-0.40	-0.43
	Park Joy	0.00	-0.14	-0.15	-0.14	-0.24
	Average	**0.00**	**-0.33**	**-0.33**	**-0.34**	**-0.40**

Table 5.13: Performance comparison in terms of motion estimation time (T_{me})

	Sequence	Schemes				
		FS [125]	**IPSO** [150]	**APSO** [152]	**PBPSO** [154]	**PMPSO**
T_{me} **(in Sec.)**	Foreman	3.59	0.25	0.25	0.33	0.31
	Highway	2.67	0.23	0.23	0.21	0.21
	Mobile	25.21	1.95	1.80	1.90	1.79
	Bus	22.10	1.91	1.93	2.03	1.77
	Crew	67.77	5.21	5.38	5.29	4.53
	Soccer	60.98	4.36	4.69	4.84	4.82
	Old Town Cross	150.47	9.87	9.76	9.49	9.51
	Park Joy	268.31	34.85	36.09	35.99	22.63
	Average	**75.14**	**7.33**	**7.52**	**7.51**	**5.69**
ΔT_{me}	Foreman	0.00	-0.93	-0.91	-0.93	-0.91
	Highway	0.00	-0.91	-0.92	-0.91	-0.92
	Mobile	0.00	-0.92	-0.92	-0.93	-0.93
	Bus	0.00	-0.91	-0.91	-0.91	-0.92
	Crew	0.00	-0.92	-0.92	-0.92	-0.93
	Soccer	0.00	-0.93	-0.92	-0.92	-0.92
	Old Town Cross	0.00	-0.93	-0.94	-0.94	-0.94
	Park Joy	0.00	-0.87	-0.87	-0.87	-0.92
	Average	**0.00**	**-0.92**	**-0.91**	**-0.92**	**-0.92**



(a) (b)

(c) (d)

(e) (f)

Figure 5.26: Subjective performance of PMPSO-ME and other existing competitive schemes in *Foreman* sequence: a) Original frame, b) FS (144.14 kbps, 38.23 dB), c) IPSO (143.75 kbps, 37.67 dB), d) APSO (143.97 kbps, 37.66 dB), e) PBPSO (143.12 kbps, 37.68 dB) and f) PMPSO (145.51 kbps, 38.39 dB)

Chapter 6

Development of Hybrid Foveated Video Compression Schemes

Preview

Now-a-days, various modes of network (wired or wireless) are existing together that support multiple data rates. The transmission of a video data with low bit-rate having good visual quality in this heterogeneous network necessitates a highly efficient video encoder. Keeping this in mind, in this chapter, we have proposed various efficient and novel encoding schemes to incorporate in existing H.264/AVC video encoder for optimizing its compression performance while maintaining high visual quality to salient regions. The comparative analysis of all proposed schemes in this thesis is presented.

The following topics are covered in this chapter.

- Introduction
- Development of hybrid foveated video compression schemes
- Comparative analysis
- Conclusion

6.1 Introduction

In a heterogeneous network, transmission of video data with uniform resolution using low data rates degrades the visual quality significantly. Therefore, to attain good visual quality, bit-rate must be increased. But, in a video frame, all regions are not equally important and hence, maintaining good visual quality to unimportant or non-salient regions comes with the cost of higher bit-rates. Based on this observation, we have proposed some foveated video compression (FVC) schemes in this doctoral research work. In a FVC, video is encoded with non-uniform resolution similar to HVS i.e., important or salient regions are encoded with sufficiently higher visual quality than their complements. To further improve the compression performance and to reduce video encoding time, we have also proposed directional transform and fast motion estimation techniques. The schemes are as follows.

1. Efficient foveated video coding scheme: Here, a video data is non-uniformly encoded to achieve higher compression ratio. The salient regions, obtained by either multi-scale phase spectrum based saliency detection (FTPBSD) or sign-DCT multi-scale pseudo-phase spectrum based saliency detection (SDCTPBSD), are encoded with higher visual quality **(Chapter 3)**.

2. Direction-adaptive fixed length directional transform scheme: A directional transform scheme based on direction-adaptive fixed length discrete cosine transform (DAFL-DCT) for intra-, and inter-frame to achieve higher coding performance in case of directional featured blocks. Two encoding modes of DAFL-DCT are proposed. One is high efficiency mode DAFL-HE and other is low complexity mode DAFL-LC **(Chapter 4)**.

3. Fast motion estimation schemes: To improve the compression gain of inter-coding and to reduce the encoding time, we have also proposed uni-modal error surface (UES) based direction-adaptive motion estimation (DAME) scheme and pattern-based modified particle swarm optimization motion estimation (PMPSO-ME) scheme based on multi-modal error surface (MES) **(Chapter 5)**.

These schemes are designed, developed and analysed independently in earlier chapters. The performance comparisons with their competitive existing schemes have been carried out with respect to various benchmark metrics along-with subjective analysis.

Now, some hybrid schemes are proposed, which are based on these schemes, to enhance the compression performance.

6.2 Development of Hybrid Foveated Video Compression Schemes

We have endeavoured to hybridize our proposed schemes, presented in Chapter-3, Chapter-4 and Chapter-5, to yield much better compression performance while retaining sufficiently high visual quality. In this regard, we have taken up three different paradigms. First, we have combined FVC algorithm with conventional DCT of H.264/AVC platform which is depicted in Figure 6.1. The second paradigm talks of associating FVC with DALF-DCT to improve the compression performance for directional featured blocks as well as to yield high visual quality promised by an FVC algorithm. This is shown in Figure 6.2. Thirdly, incorporating a fast motion estimation algorithm to the second paradigm yields a totally different platform, which is depicted in Figure 6.3. The paradigm is expected to yield the best performance, in terms of compression performance and visual quality as well, since all three independent modules of our research are fused together.

These three paradigms will produce various algorithms. From intuition, it is understood that all the three paradigms will not yield the same quality. In fact, they are not expected to.

Accordingly, following hybrid schemes are proposed .

(1) Paradigm-I

(i) **FVC-FTPBSD-DCT** scheme: —a combination of FTPBSD based FVC and conventional DCT schemes;

(ii) **FVC-SDCTPBSD-DCT** scheme: —a combination of SDCTPBSD based FVC and conventional DCT schemes.

(2) Paradigm-II

(i) **FVC-FTPBSD-DAFL-HE** scheme: —a combination of FTPBSD based FVC and DAFL-HE schemes;

(ii) **FVC-FTPBSD-DAFL-LC** scheme: —a combination of FTPBSD based FVC and DAFL-LC schemes;

(iii) **FVC-SDCTPBSD-DAFL-HE** scheme: —a combination of SDCTPBSD based FVC and DAFL-HE schemes;

(iv) **FVC-SDCTPBSD-DAFL-LC** scheme: —a combination of SDCTPBSD based FVC and DAFL-LC schemes;

(3) Paradigm-III

(i) **FVC-FTPBSD-DAFL-HE-DAME** scheme: —a combination of FTPBSD based FVC, DAFL-HE and DAME schemes;

(ii) **FVC-FTPBSD-DAFL-HE-PMPSO** scheme: —a combination of FTPBSD based FVC, DAFL-HE and PMPSO-ME schemes;

(iii) **FVC-FTPBSD-DAFL-LC-DAME** scheme: —a combination of FTPBSD based FVC, DAFL-LC and DAME schemes;

(iv) **FVC-FTPBSD-DAFL-HE-PMPSO** scheme: —a combination of FTPBSD based FVC, DAFL-LC and PMPSO-ME schemes.

(v) **FVC-SDCTPBSD-DAFL-HE-DAME** scheme: —a combination of SDCTPBSD based FVC, DAFL-HE and DAME schemes;

(vi) **FVC-SDCTPBSD-DAFL-HE-PMPSO** scheme: —a combination of SDCTPBSD based FVC, DAFL-HE and PMPSO-ME schemes;

(vii) **FVC-SDCTPBSD-DAFL-LC-DAME** scheme: —a combination of SDCTPBSD based FVC, DAFL-LC and DAME schemes;

(viii) **FVC-SDCTPBSD-DAFL-HE-PMPSO** scheme: —a combination of SDCTPBSD based FVC, DAFL-LC and PMPSO-ME schemes.

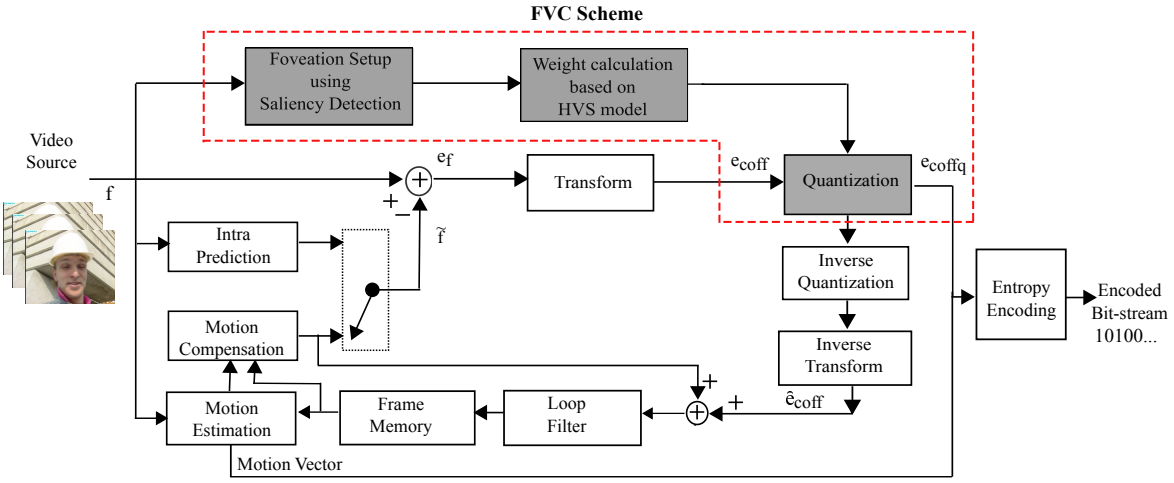


Figure 6.1: Block diagram of Paradigm-I (FVC with conventional DCT of H.264/AVC)

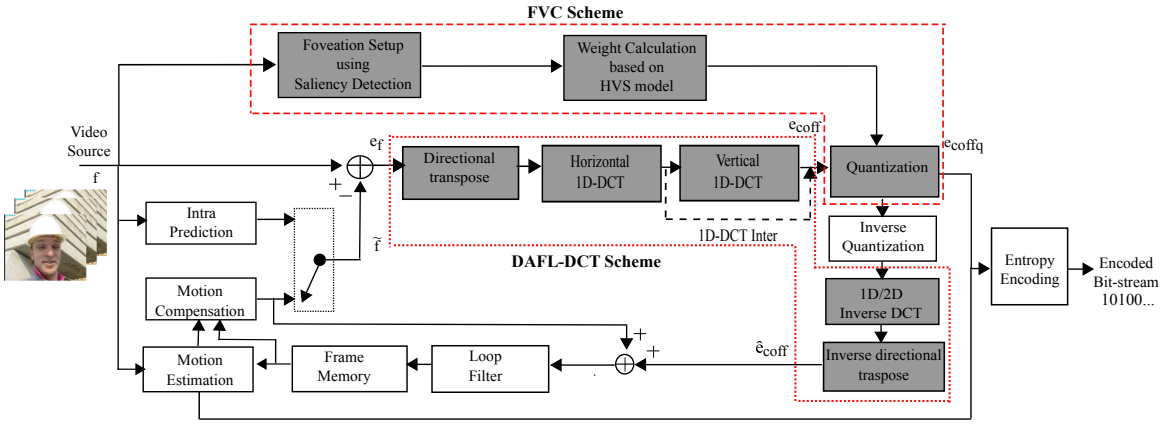


Figure 6.2: Block diagram of Paradigm-II (FVC with DAFL-DCT)

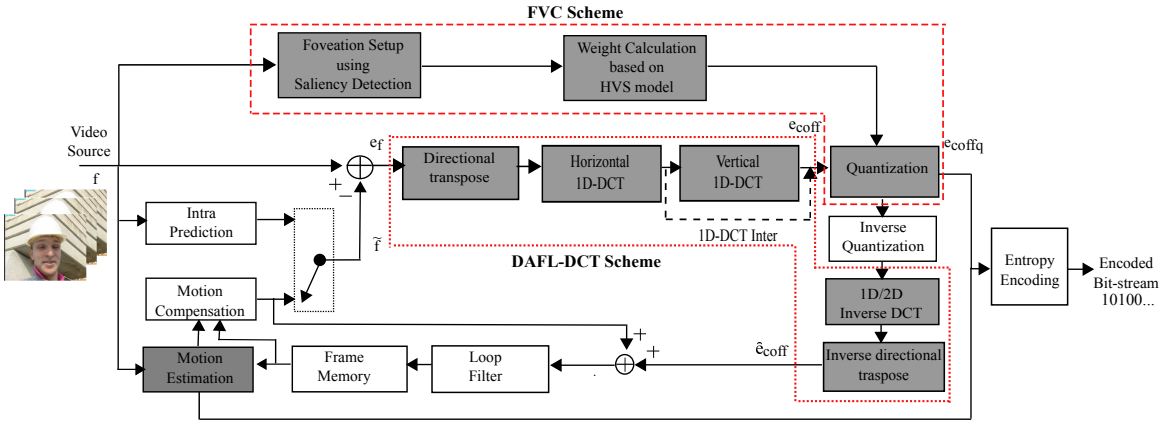


Figure 6.3: Block diagram of Paradigm-III (FVC alongwith DAFL-DCT and ME schemes)

6.3 Comparative Analysis

The hybrid schemes are incorporated into H.264/AVC platform and experiments have been carried out on standard video sequences: *Foreman, Highway, Mobile, Bus, Crew, Soccer, Old Town Cross* and *Park Joy*. The performance analysis with respect to BD-PSNR,

Table 6.1: Comparative Bjontegaard metric[36] performance analysis of FTPBSD based FVC schemes in H.264/AVC platform

	Schemes	Sequence								
		Foreman	Highway	Mobile	Bus	Crew	Soccer	Old Town Cross	Park Joy	Average
BD-PSNR (in dB)	FVC-FTPBSD-DCT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	FVC-FTPBSD-DAFL-HE	0.34	0.27	0.47	0.41	0.25	0.26	0.08	0.54	**0.33**
	FVC-FTPBSD-DAFL-LC	0.19	0.19	0.33	0.20	0.17	0.16	0.09	0.23	0.20
	FVC-FTPBSD-DAFL-HE-DAME	0.35	0.18	0.65	-0.07	1.39	-0.18	-0.01	0.00	0.29
	FVC-FTPBSD-DAFL-HE-PMPSO	0.47	0.16	0.55	-0.03	1.13	-0.10	-0.03	-0.43	0.22
	FVC-FTPBSD-DAFL-LC-DAME	0.14	0.10	0.51	-0.22	1.02	-0.26	0.06	0.10	0.18
	FVC-FTPBSD-DAFL-LC-PMPSO	0.29	0.08	0.40	0.04	0.83	-0.09	0.03	-0.17	0.18
BD-SSIM	FVC-FTPBSD-DCT	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	FVC-FTPBSD-DAFL-HE	0.0034	0.0038	0.0090	0.0060	0.0027	0.0010	-0.0024	0.0137	**0.0046**
	FVC-FTPBSD-DAFL-LC	0.0016	0.0022	0.0067	0.0025	0.0012	0.0003	-0.0010	0.0067	0.0025
	FVC-FTPBSD-DAFL-HE-DAME	0.0030	0.0029	0.0116	-0.0285	0.0293	-0.0172	-0.0065	-0.0085	-0.0017
	FVC-FTPBSD-DAFL-HE-PMPSO	0.0052	0.0024	0.0099	-0.0143	0.0230	-0.0078	-0.0058	-0.0203	-0.0010
	FVC-FTPBSD-DAFL-LC-DAME	-0.0001	0.0012	0.0092	-0.0311	0.0221	-0.0188	-0.0032	-0.0121	-0.0041
	FVC-FTPBSD-DAFL-LC-PMPSO	0.0025	0.0008	0.0076	-0.0175	0.0176	-0.0106	-0.0028	-0.0188	-0.0026
BD-bitrate (in %)	FVC-FTPBSD-DCT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	FVC-FTPBSD-DAFL-HE	-8.41	-16.31	-14.55	-11.17	-6.42	-9.85	-13.46	-13.57	**-11.72**
	FVC-FTPBSD-DAFL-LC	-4.87	-11.52	-10.46	-5.22	-3.99	-5.81	-9.28	-4.47	-6.95
	FVC-FTPBSD-DAFL-HE-DAME	-8.63	-11.83	-19.33	1.12	-26.44	3.86	-19.20	-0.96	-10.18
	FVC-FTPBSD-DAFL-HE-PMPSO	-11.35	-11.32	-16.50	0.08	-22.81	1.34	-15.76	2.65	-9.21
	FVC-FTPBSD-DAFL-LC-DAME	-3.67	-6.73	-15.52	4.33	-21.39	3.93	-14.61	-1.28	-6.87
	FVC-FTPBSD-DAFL-LC-PMPSO	-7.21	-5.82	-12.46	-0.94	-18.17	2.43	-11.06	6.32	-5.87

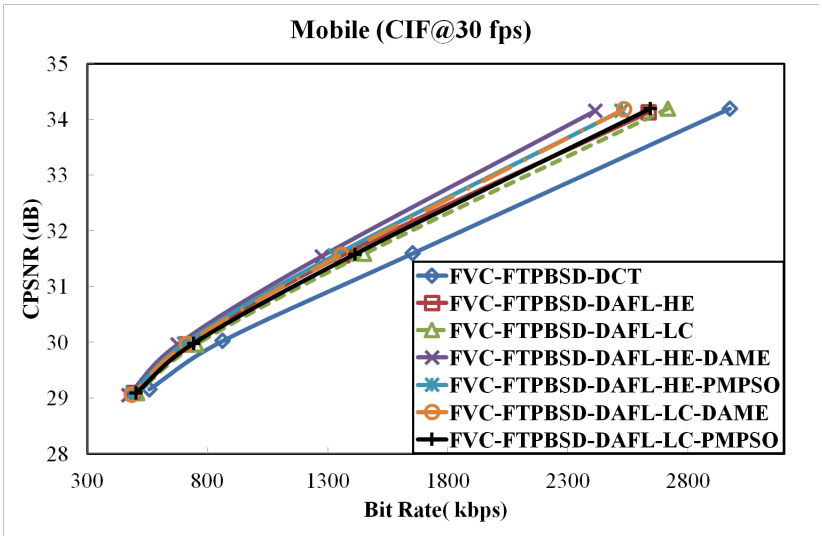
Table 6.2: Comparative Bjontegaard metric[36] performance analysis of SDCTPBSD based FVC schemes in H.264/AVC platform

	Schemes	Sequence								
		Foreman	Highway	Mobile	Bus	Crew	Soccer	Old Town Cross	Park Joy	Average
BD-PSNR (in dB)	FVC-SDCTPBSD-DCT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	FVC-SDCTPBSD-DAFL-HE	0.39	0.23	0.52	0.30	0.20	0.27	0.17	0.27	0.30
	FVC-SDCTPBSD-DAFL-LC	0.18	0.16	0.39	0.12	-0.03	0.20	0.12	0.27	0.18
	FVC-SDCTPBSD-DAFL-HE-DAME	0.46	0.13	0.77	-0.02	1.03	-0.09	0.24	0.16	**0.34**
	FVC-SDCTPBSD-DAFL-HE-PMPSO	0.62	0.11	0.67	0.10	0.88	-0.05	0.18	-0.04	0.31
	FVC-SDCTPBSD-DAFL-LC-DAME	0.26	0.04	0.61	-0.13	0.80	-0.08	0.21	0.14	0.23
	FVC-SDCTPBSD-DAFL-LC-PMPSO	0.39	0.03	0.51	0.02	0.60	-0.02	0.14	-0.05	0.20
BD-SSIM	FVC-SDCTPBSD-DCT	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	FVC-SDCTPBSD-DAFL-HE	0.0050	0.0032	0.0104	0.0031	0.0011	0.0014	0.0015	0.0046	**0.0038**
	FVC-SDCTPBSD-DAFL-LC	0.0021	0.0018	0.0085	0.0000	-0.0019	0.0007	0.0011	0.0049	0.0022
	FVC-SDCTPBSD-DAFL-HE-DAME	0.0055	0.0019	0.0153	-0.0301	0.0179	-0.0161	0.0023	-0.0173	-0.0026
	FVC-SDCTPBSD-DAFL-HE-PMPSO	0.0075	0.0017	0.0137	-0.0184	0.0145	-0.0077	0.0016	-0.0230	-0.0013
	FVC-SDCTPBSD-DAFL-LC-DAME	0.0024	0.0001	0.0123	-0.0333	0.0137	-0.0168	0.0022	-0.0180	-0.0047
	FVC-SDCTPBSD-DAFL-LC-PMPSO	0.0044	0.0000	0.0106	-0.0217	0.0100	-0.0098	0.0014	-0.0259	-0.0039
BD-bitrate (in %)	FVC-SDCTPBSD-DCT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	FVC-SDCTPBSD-DAFL-HE	-8.78	-11.58	-18.55	-8.71	-5.35	-8.13	-10.07	-5.97	-9.64
	FVC-SDCTPBSD-DAFL-LC	-4.22	-8.27	-13.45	-3.24	0.40	-6.14	-6.61	-7.89	-6.18
	FVC-SDCTPBSD-DAFL-HE-DAME	-10.27	-7.17	-24.39	-0.90	-23.13	2.43	-15.77	-2.77	**-10.24**
	FVC-SDCTPBSD-DAFL-HE-PMPSO	-13.16	-6.66	-21.47	-1.65	-20.06	1.13	-12.36	3.17	-8.88
	FVC-SDCTPBSD-DAFL-LC-DAME	-5.84	-2.45	-19.74	1.67	-18.37	2.75	-12.34	-3.52	-7.23
	FVC-SDCTPBSD-DAFL-LC-PMPSO	-8.76	-2.18	-17.06	1.46	-14.41	0.69	-8.78	1.72	-5.92

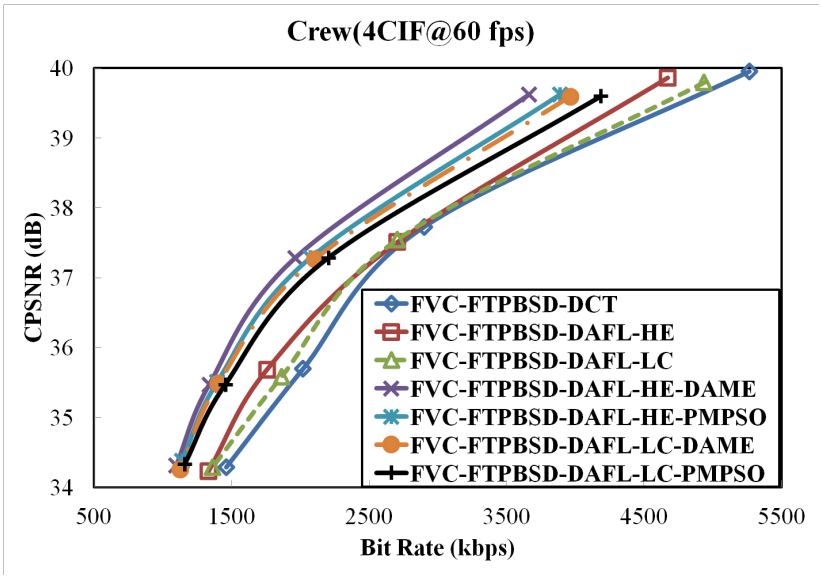
BD-SSIM and BD-bitrate [36], rate-distortion (R-D) curves, encoding time complexity and subjective quality have been made to derive an overall conclusion.

6.3.1 Experiment 1: Bjontegaard metrics performance

In this experiment, each set of the proposed hybrid FVC schemes are compared with proposed conventional FVC schemes (FVC-FTPBSD-DCT and FVC-SDCTPBSD-DCT)



(a)

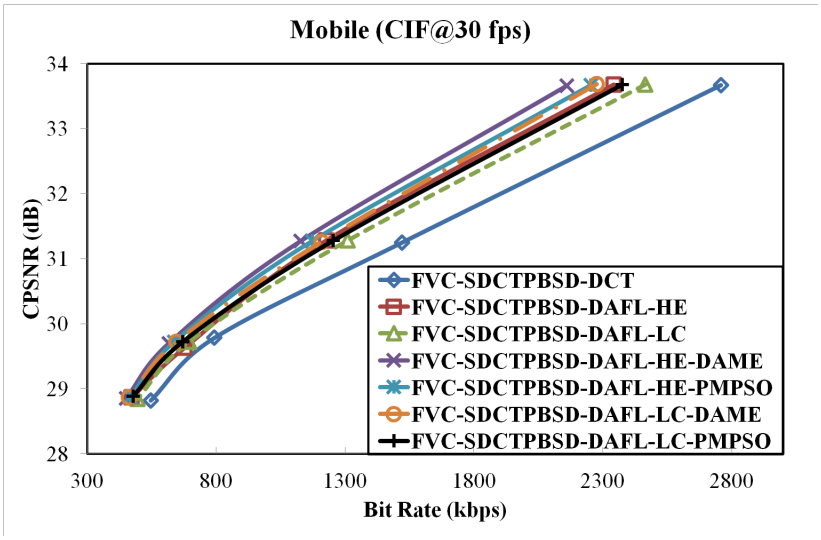


(b)

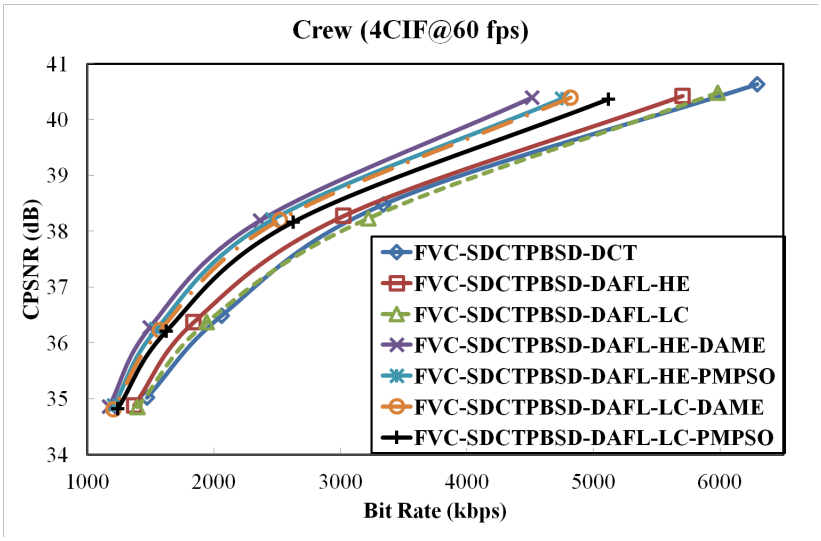
Figure 6.4: Rate-distortion curves for FTPBSD based hybrid schemes

respectively. The performance values in terms of BD-PSNR, BD-SSIM and BD-bitrate are summarized in Table 6.1 and Table 6.2. In Bjontegaard metric positive numbers in BD-PSNR and BD-SSIM represent gain, while negative numbers in BD-bitrate show reduction in bit-rate. The R-D curves for *Mobile* and *Crew* sequences are presented in Figure 6.4 and Figure 6.5 for FTPBSD based FVC schemes and SDCTPBSD based FVC schemes respectively.

From Table 6.1, it is observed that FVC-FTPBSD-DAFL-HE yields improvement in BD-PSNR of 0.33 dB (or equivalently improvement in BD-SSIM of 0.0046 or 11.72% reduction in BD-bitrate) on average with respect to FVC-FTPBSD-DCT. It means FVC-FTPBSD-DAFL-HE outperforms other hybrid schemes and yields better PSNR gain and visual quality for lower bit-rate. It is also found that FVC-FTPBSD-DAFL-HE-DAME



(a)



(b)

Figure 6.5: Rate-distortion curves for SDCTPBSD based hybrid schemes

achieves a quite noticeable improvement in BD-PSNR of 1.39 dB (or equivalently improvement in BD-SSIM of 0.0293 or 26.44% reduction in BD-bitrate) for *Crew* video sequence.

From Table 6.2, it is observed that FVC-SDCTPBSD-DAFL-HE-DAME yields improvement in BD-PSNR of 0.34 dB (or equivalently reduction in BD-SSIM of 0.0026 or 10.24% reduction in BD-bitrate) on average with respect to FVC-SDCTPBSD-DCT. It means FVC-SDCTPBSD-DAFL-HE-DAME outperforms other hybrid schemes and yields better PSNR gain for lower bit-rate but marginally reduced visual quality. It is also found that FVC-FTPBSD-DAFL-HE-DAME achieves a quite noticeable improvement in BD-PSNR of 1.03 dB (or equivalently improvement in BD-SSIM of 0.0179 or 23.13% reduction in BD-bitrate) for *Crew* video sequence.

The R-D curves, from Figure 6.4 and Figure 6.5 as well as Bjontegaard

Table 6.3: Comparative ΔT encoding time analysis of proposed hybrid FVC schemes in H.264/AVC platform

Schemes	Sequence								
	Foreman	Highway	Mobile	Bus	Crew	Soccer	Old Town Cross	Park Joy	Average
FVC-FTPBSD-DCT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FVC-FTPBSD-DAFL-HE	-0.37	-0.34	-0.34	-0.37	-0.44	-0.39	-0.36	-0.36	-0.37
FVC-FTPBSD-DAFL-LC	-0.63	-0.57	-0.60	-0.61	-0.65	-0.62	-0.61	-0.58	-0.61
FVC-FTPBSD-DAFL-HE-DAME	-0.43	-0.39	-0.42	-0.32	-0.50	-0.40	-0.44	-0.35	-0.41
FVC-FTPBSD-DAFL-HE-PMPSO	-0.43	-0.40	-0.42	-0.37	-0.50	-0.44	-0.44	-0.38	-0.42
FVC-FTPBSD-DAFL-LC-DAME	-0.69	-0.63	-0.68	-0.61	-0.72	-0.66	-0.68	-0.62	-0.66
FVC-FTPBSD-DAFL-LC-PMPSO	-0.70	-0.63	-0.68	-0.65	-0.73	-0.67	-0.68	-0.63	-0.67
FVC-SDCTPBSD-DCT	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
FVC-SDCTPBSD-DAFL-HE	-0.37	-0.33	-0.34	-0.36	-0.43	-0.39	-0.35	-0.35	-0.37
FVC-SDCTPBSD-DAFL-LC	-0.62	-0.56	-0.60	-0.61	-0.65	-0.62	-0.60	-0.59	-0.61
FVC-SDCTPBSD-DAFL-HE-DAME	-0.42	-0.37	-0.42	-0.31	-0.49	-0.42	-0.42	-0.34	-0.40
FVC-SDCTPBSD-DAFL-HE-PMPSO	-0.43	-0.38	-0.43	-0.37	-0.50	-0.44	-0.42	-0.37	-0.42
FVC-SDCTPBSD-DAFL-LC-DAME	-0.69	-0.61	-0.68	-0.61	-0.72	-0.66	-0.67	-0.61	-0.66
FVC-SDCTPBSD-DAFL-LC-PMPSO	-0.69	-0.61	-0.68	-0.64	-0.72	-0.68	-0.67	-0.63	-0.67

metrics, from Table 6.1 and Table 6.2, signify the superior performance of FVC-FTPBSD-DAFL-HE-DAME and FVC-SDCTPBSD-DAFL-HE-DAME hybrid schemes compared to other hybrid schemes.

6.3.2 Experiment 2: Analysis of encoding time complexity

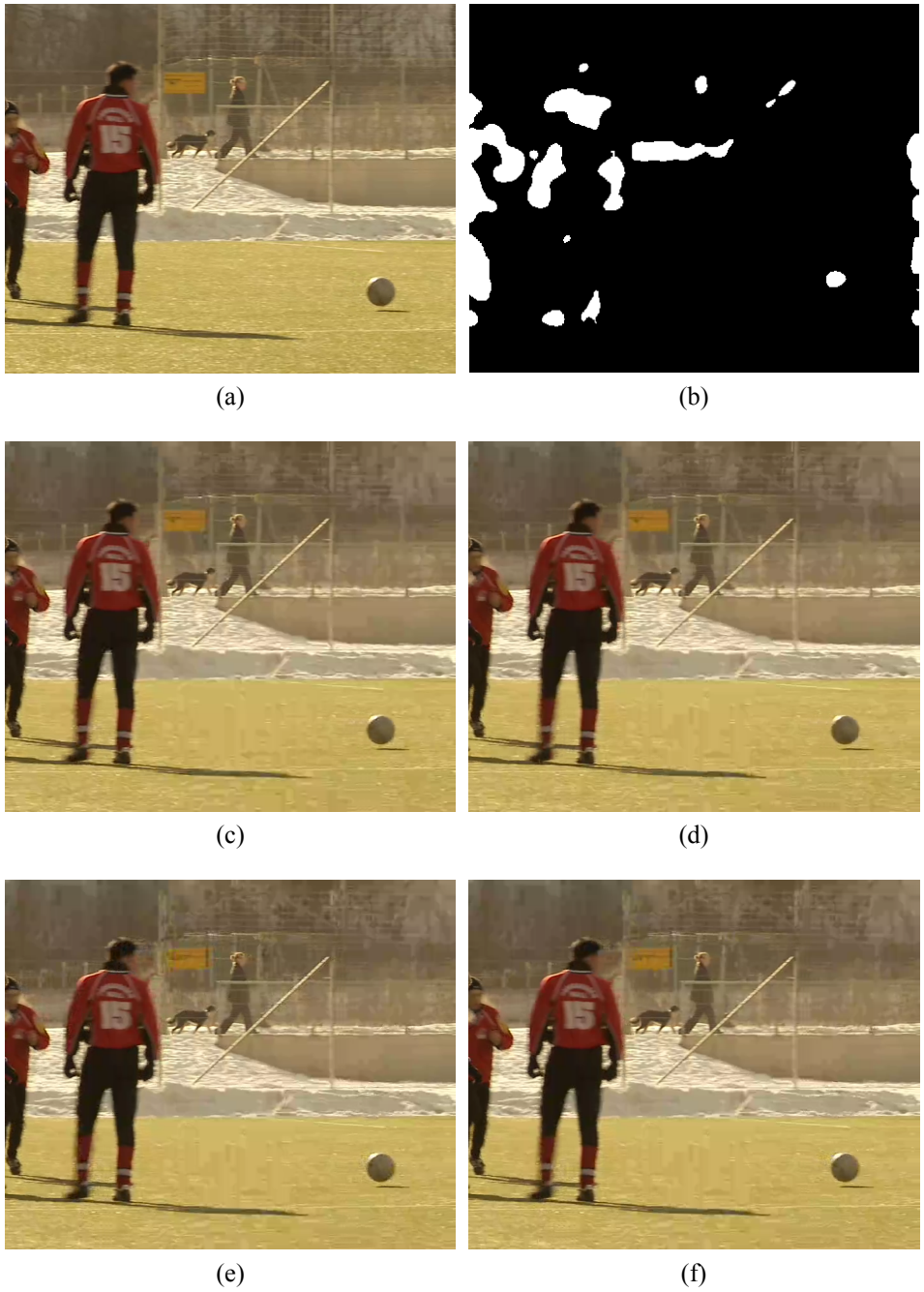
The encoding time complexity in terms of ΔT, defined in (3.20), is specified in Table 6.3. The ΔT represents relative change in encoding time and hence positive value indicates increase in encoding time and vice-versa. From Table 6.3, it is observed that DAFL-LC based hybrid schemes outperform DAFL-HE based hybrid schemes meeting its design goal. Similarly, PMPSO-ME based hybrid schemes are faster than DAME based hybrid schemes but with a compromise with the objective performance.

6.3.3 Experiment 3: Subjective evaluation

The comparative analysis in terms of subjective evaluation of FTPBSD based hybrid FVC schemes are shown in Figure 6.6 for *Soccer* sequence along-with the FTPBSD object map for the corresponding frame. Similarly, SDCTPBSD based hybrid FVC schemes are shown in Figure 6.7. It is clearly seen that in FVC schemes salient regions are encoded with higher bit-rate as compared to non-salient regions.

6.4 Conclusion

Various hybrid foveated video compression schemes are generated from different combinations of proposed FVC schemes (FTPBSD based FVC scheme and SDCTPBSD based FVC scheme), directional transform and motion estimation schemes. The proposed schemes are meticulously evaluated through various experiments. Among all hybrid schemes, FVC-SDCT-DAFL-HE-DAME shows a superior performance in terms of



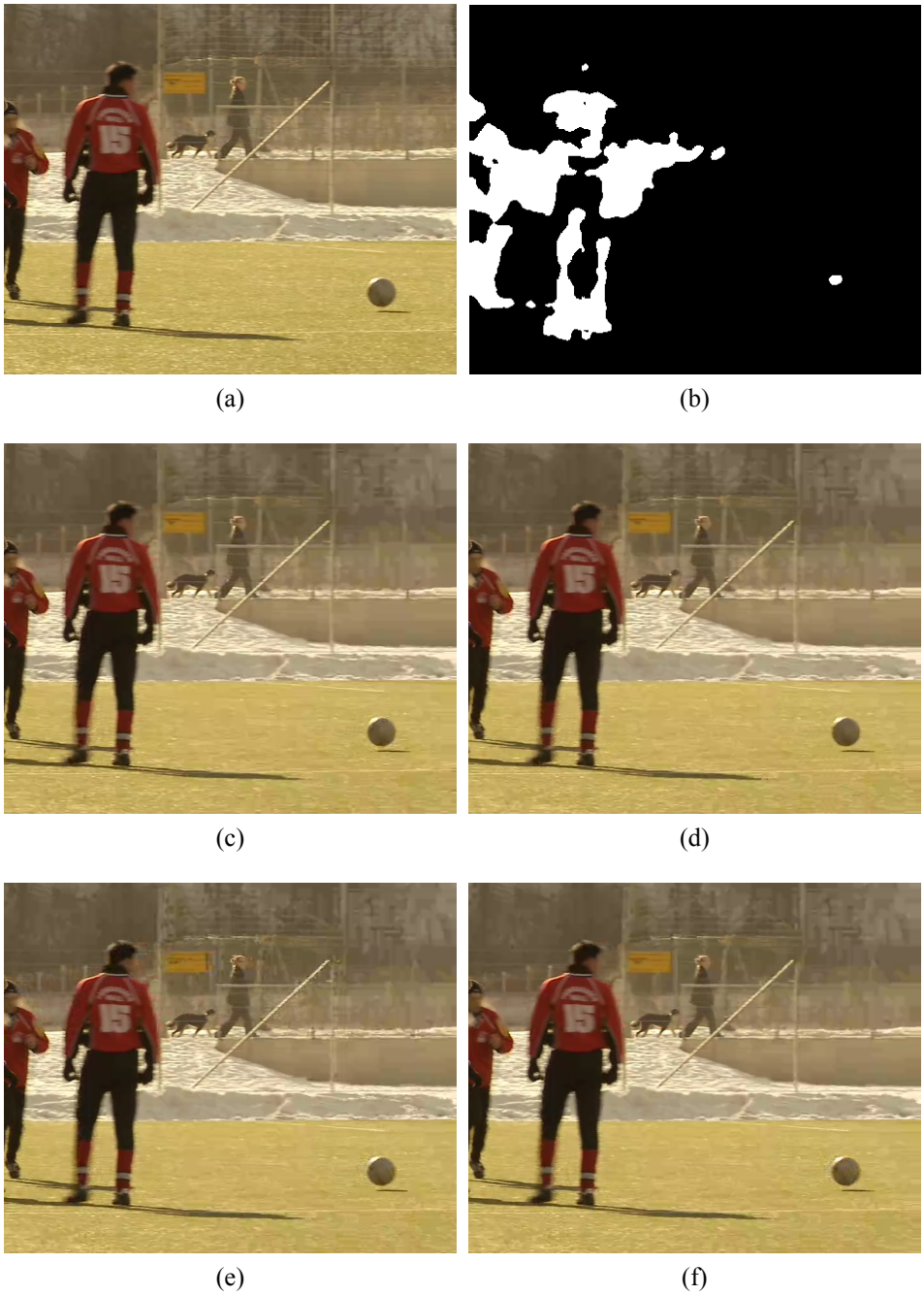
(a) (b)

(c) (d)

(e) (f)

Figure 6.6: Comparative subjective evaluation of reconstructed frame for FTPBSD based hybrid FVC schemes with QP = 32 in *Soccer* sequence: a) Conventional encoder, b) FTPBSD object map, c) FVC-FTPBSD-DCT, d) FVC-FTPBSD-DAFL-HE, e) FVC-FTPBSD-DAFL-HE-DAME and f) FVC-FTPBSD-DAFL-HE-PMPSO

objective and subjective evaluations. The proposed schemes are suitable for various H.264/AVC platform based low bit-rate applications like mobile based video telephony and conferencing as well as medium bit-rate applications such as standard-definition TV broadcasting and web based video related services.



(a) (b)

(c) (d)

(e) (f)

Figure 6.7: Comparative subjective evaluation of reconstructed frame for SDCTPBSD based hybrid FVC schemes with QP = 32 in *Soccer* sequence: a) Conventional encoder, b) SDCTPBSD object map, c) FVC-SDCTPBSD-DCT, d) FVC-SDCTPBSD-DAFL-HE, e) FVC-SDCTPBSD-DAFL-HE-DAME and f) FVC-SDCTPBSD-DAFL-HE-PMPSO

Chapter 7

Conclusion

Preview

The schemes, proposed in this thesis, have been developed for providing fast and efficient foveated video compression that yield higher compression performance with lower computational cost and sufficiently higher visual quality at salient regions as well for H.264/AVC platform. In this chapter, the overall conclusions are presented and the contributions are summarised.

The following topics are covered in this chapter.

- Performance analysis
- Conclusion
- Scope for future work

7.1 Performance Analysis

The efficient foveated video compression schemes using saliency map (FVC-FTPBSD and FVC-SDCTPBSD) for H.264/AVC platform are proposed in Chapter 3. The saliency maps are evaluated using proposed FTPBSD and SDCTPBSD saliency detection techniques. The SDCTPBSD yields much higher precision, recall and AUC than FTPBSD. It has been observed with experimental results that the proposed FVC schemes (FVC-FTPBSD and FVC-SDCTPBSD) greatly reduce the bit-rate of a video data while retaining high visual quality to its salient regions.

To further optimize the performance of H.264/AVC video coding, an efficient direction-adaptive fixed length discrete cosine transform (DAFL-DCT) for directional featured blocks is proposed in Chapter 4. The DAFL-DCT proposes two sets of eight directional transform modes for 4×4 and 8×8 blocks, one for each. In intra-frame coding, 2D-DAFL-DCTs are used, whereas conventional 2D-DCT and 1D-DAFL-DCTs are adaptively chosen in inter-frame encoding for each block. Two encoding modes of DAFL-DCT: DAFL-HE and DAFL-LC are proposed. The DAFL-HE is a high efficiency mode that selects an optimum DAFL-DCT transform mode for each block and yields superior

Table 7.1: Comparative compression performance of the proposed hybrid foveated video compression schemes for *Old town cross* (HD720p) video sequence for H.264/AVC platform

Schemes	**CPSNR (dB)**	**MSSIM**	**Bit-Rate (kbps)**	**Encoding Time (Seconds)**
H.264-DCT	39.25	0.9275	17872.26	1490.96
FVC-FTPBSD-DCT	**37.16**	0.9039	5647.09	900.44
FVC-FTPBSD-DAFL-HE	37.09	0.9027	4729.93	572.88
FVC-FTPBSD-DAFL-LC	37.11	0.9030	5076.22	355.11
FVC-FTPBSD-DAFL-HE-DAME	37.04	0.9024	**4380.54**	506.02
FVC-FTPBSD-DAFL-HE-PMPSO	37.06	0.9027	4572.08	504.30
FVC-FTPBSD-DAFL-LC-DAME	37.06	0.9024	4692.13	284.50
FVC-FTPBSD-DAFL-LC-PMPSO	37.08	0.9028	4891.89	**284.20**
H.264-DCT	39.25	0.9275	17872.26	1490.96
FVC-SDCTPBSD-DCT	**37.29**	0.9060	6071.88	882.65
FVC-SDCTPBSD-DAFL-HE	37.22	0.9049	5286.67	575.96
FVC-SDCTPBSD-DAFL-LC	37.24	0.9052	5585.09	355.84
FVC-SDCTPBSD-DAFL-HE-DAME	37.18	0.9047	**4895.12**	510.01
FVC-SDCTPBSD-DAFL-HE-PMPSO	37.19	0.9049	5109.91	508.33
FVC-SDCTPBSD-DAFL-LC-DAME	37.20	0.9048	5182.95	**287.03**
FVC-SDCTPBSD-DAFL-LC-PMPSO	37.22	0.9051	5408.69	287.17

compression performance at the cost of higher complexity as compared to low complexity mode represented as DAFL-LC. The proposed DAFL-DCT is shown to have superior performance than the conventional 2D-DCT and other existing directional transforms in terms of both objective and subjective analysis.

In Chapter 5, two fast and efficient BMME schemes are proposed to exploit temporal correlation between video frames. One is UES based direction-adaptive motion estimation (DAME) scheme and another is pattern-based modified particle swarm optimization motion estimation (PMPSO-ME) based on MES. The DAME provides slightly better results as compared to PMPSO-ME.

To have a bird's eye view on performance of all the proposed hybrid FVC schemes, their results, in terms of CPSNR, MSSIM, bit-rate and encoding time, are presented in Table 7.1 for *Old town cross* (HD720p) test video sequence for H.264/AVC platform.

From Table 7.1, it is observed that FVC-FTPBSD and FVC-SDCTPBSD yield reduction in CPSNR by 2.09 dB and 1.96 dB on average and also exhibit reduction in bit-rate by 68.40% and 66.03% on average as compared to conventional H.264-DCT scheme, respectively. Of course, visual quality of FVC-SDCTPBSD is superior to that of FVC-FTPBSD as indicated by MSSIM value.The proposed DAFL-HE further reduces the bit-rate while maintaining the visual quality and outperforms DAFL-LC. The proposed scheme, PMPSO-ME reduces encoding time and shows better performance in terms of CPSNR and MSSIM as compared to DAME scheme but yields higher bit-rate. Thus, for a similar bit-rate, CPSNR and

MSSIM value of PMPSO-ME will be less than that provided by DAME. Hence, it can be concluded that DAME outperforms PMPSO-ME. A combination of DAFL-HE and DAME with FVC schemes improves the compression performance and retains visual quality than other combinations.

7.2 Conclusion

The analysis, presented in the previous section, leads us to draw the following conclusion.

- SDCTPBSD based FVC schemes exhibit better CPSNR and similar MSSIM value to FTPBSD based FVC schemes, but provide a slightly less compression ratio.
- The DAFL-HE yields promising results, maintaining the quality in terms of objective metrics with slightly extra encoding time as compared to DAFL-LC.
- The DAME scheme gives better CPSNR, MSSIM value and reduction in bit-rate compared to PMPSO-ME, but marginally slower in encoding.
- The FVC-SDCTPBSD-DAFL-HE-DAME scheme outperforms competitive FVC-SDCTPBSD-DAFL-HE-PMPSO-ME scheme and yields better CPSNR, MSSIM value and reduction in bit-rate.

Finally, it may be concluded that among the hybrid schemes, **FVC-SDCTPBSD-DAFL-HE-DAME** has a superior performance in all benchmark metrics. In general, it is observed that each proposed scheme shows superior performance as compared to competitive schemes existing in literature. The proposed schemes are suitable for various H.264/AVC platform based low bit-rate applications like mobile based video telephony and conferencing as well as medium bit-rate applications such as standard-definition TV broadcasting and web based video related services.

7.3 Scope For Future Work

There is sufficient scope to carry out further research in developing efficient video compression schemes employing the following techniques.

(a) Neural network (NN) and fuzzy inference system (FIS) are very good adaptive systems. Much more research is expected to fine tune the transform and motion estimation modules using on-line or off-line training modules.

(b) Efficient hardware, for video codec, may be designed employing pipelined VLSI architecture.

(c) Intra-prediction, loop filter and entropy encoding modules play important roles in improving compression ratio and visual quality in H.264/AVC. Many fast and efficient techniques for these modules are expected in near future.

(d) A challenging task may be taken up to modify the proposed schemes for making them compatible with the latest coding standard, H.265/HEVC.

References

[1] I. Richardson, *Video Codec Design: Developing Image and Video Compression Systems*, 1st ed. John Wiley & Sons, Ltd, 2002.

[2] J. D. Gibson and A. Bovik, Eds., *Handbook of Image and Video Processing*, 1st ed. Orlando, FL, USA: Academic Press, Inc., 2000.

[3] K. Jack, *Video Demystified: A Handbook for the Digital Engineer*, 5th ed. Newton, MA, USA: Newnes, 2007.

[4] R. W. G. Hunt and P. M. R., *Measuring Color*, 4, Ed. John Wiley & Sons Inc., September 2011.

[5] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[6] ITU-T Recommendation H.264 / ISO/IEC 14496-10, *Advanced Video Coding for Generic Audiovisual Services*, ITU-T / ISO/IEC Std., March 2005.

[7] Recommendation ITU-R BT.601-5, *Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios,*, ITU-T Std., 1995.

[8] L. Hanzo, P. Cherriman, and J. Streit, *Video Compression and Communications: From Basics to H.261, H.263, H.264, MPEG4 for DVB and HSDPA-Style Adaptive Turbo-Transceivers*, 2nd ed. Wiley-IEEE Press, 2007.

[9] S. B. Solak and F. Labeau, *Sustainable ICTs and Management Systems for Green Computing*. IGI Global, June 2012, ch. Green Video Compression for Portable and Low-Power Applications, pp. 325–349.

[10] A. Malewar, A. Bahadarpurkar, and V. Gadre, "A linear rate control model for better target buffer level tracking in H.264," *Signal, Image and Video Processing*, vol. 7, pp. 275–286, 2011.

[11] "International telecommunication union-telecommunication," http://www.itu.int.

[12] "International organization for standardization," http://www.iso.org.

[13] ITU-T Recommendation H.261, *Video codec for audiovisual services at p X 64 kbit/s*, ITU-T Std., December 1990.

[14] ISO/IEC Standard 11172-2, *Information technology: coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s-part 2: Video*, ISO/IEC Std., 1993.

[15] ISO/IEC Standard 13818-2, *Information technology: generic coding of moving pictures and associated audio information: Video*, ISO/IEC Std., 1995.

[16] ITU-T Recommendataion H.262, *Information technology - Generic coding of moving pictures and associated audio information: Video*, ITU-T Std., July 1995.

[17] ITU-T Recommendataion H.263, *Video coding for low bit rate communication*, ITU-T Std., 03 1996.

[18] M. G. Strinzis, "Object-based coding of stereospic and 3D image sequences," *IEEE Signal Process. Mag.*, pp. 14–28, 1999.

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

[20] ISO/IEC Standard 14996-2, *Information technology: coding of audio-visual objects-part 2: Visual*, ISO/IEC Std., 1998.

[21] ITU-T Recommendation H.265 / ISO/IEC 23008-2, *High Efficiency Video Coding (HEVC)*, ITU-T / ISO/IEC Std., October 2014.

[22] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec 2012.

[23] J. Vanne, M. Viitanen, T. D. Hamalainen, and A. Hallapuro, "Comparative rate-distortion-complexity analysis of HEVC and AVC video codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1885–1898, Dec 2012.

[24] M. B. Dissanayake and D. L. B. Abeyrathna, "Performance comparison of HEVC and H.264/AVC standards in broadcasting environments," *Information Processing Systems*, vol. 11, no. 3, pp. 483–494, September 2015.

[25] I. E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia*. John Wiley & Sons, 2003.

[26] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits Syst. Mag.*, vol. 4, no. 1, pp. 7–28, First 2004.

[27] X. Tian, T. M. Le, and Y. Lian, *Entropy Coders of the H.264/AVC Standard*, ser. Signals and Communication Technology. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ch. Review of CAVLC, Arithmetic Coding, and CABAC, pp. 29–39.

[28] P. G. Barten, *Contrast sensitivity of the human eye and its effects on image quality*. SPIE press, 1999, vol. 72.

[29] M. Vranješ, S. Rimac-Drlje, and D. Žagar, "Objective video quality metrics," in *49th Int. Symposium ELMAR-2007 focused on Mobile Multimedia*, 2007.

[30] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.

[31] A. H. Sadka, *Compressed video communications*. Chichester, England: John Wiley & Sons, 2002.

[32] H. Chen, M. Sun, and E. Steinbach, "Low-complexity bayer-pattern video compression using distributed video coding," in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009, pp. 725 715–725 715.

[33] W. Zhou, L. Liang, and A. Bovik, "Video quality assessment using structural distortion measurement," in *Proc. Int. Conf. on Image Processing*, vol. 3, Rochester, NY, USA, 2002.

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

[35] R. Dosselmann and X. Yang, "A comprehensive assessment of the structural similarity index," *Signal, Image and Video Processing*, vol. 5, no. 1, pp. 81–91, 2011.

[36] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," *VCEG-M33, SG16 (VCEG)*, pp. 1–4, April 2001.

[37] J. K. T. Neil D. B. Bruce, "Saliency based on information maximization," *In Advances in Neural Information Processing Systems (NIPS)*, vol. 18, pp. 155–162, 2005.

[38] L. K. Westin, “Receiver operating characteristic (ROC) analysis evaluating discriminance effects among decision support systems,” Umea University, Sweden, Tech. Rep., Department of Computing Science 2004.

[39] E. C. Reed and F. Dufaux, “Constrained bit-rate control for very low bit-rate streaming-video applications,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 7, pp. 882–889, July 2001.

[40] Z. Wen, Z. Liu, M. Cohen, J. Li, K. Zheng, and T. Huang, “Low bit-rate video streaming for face-to-face teleconference,” in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME ’04)*, vol. 3, 2004, pp. 1631–1634.

[41] P. Waingankar and G. S. Hayagreev, “Efficient low bit rate video compression technique for mobile applications,” in *Proc. Int. Conf. and Workshop on Emerging Trends in Technology (ICWET 2011)*, TCET, Mumbai, India, 2011, pp. 109–112.

[42] R. Rosenbaum and H. Schumann, “On-demand foveation for motion-JPEG2000 encoded imagery,” in *Int. Symposium on Communications and Information Technologies (ISCIT ’07)*, Oct 2007, pp. 456–461.

[43] J. C. Galan-Hernandez, V. Alarcon-Aquino, O. Starostenko, and J. M. Ramirez-Cortes, “Wavelet-based foveated compression algorithm for real-time video processing,” in *Proc. IEEE Conf. Electronics, Robotics and Automotive Mechanics (CERMA ’10)*, Washington, DC, USA, 2010, pp. 405–410.

[44] J. Ryoo, K. Yun, D. Samaras, S. R. Das, and G. J. Zelinsky, “Design and evaluation of a foveated video streaming service for commodity client devices,” in *Proc. Conf. ACM Multimedia Systems (MMSys)*, Klagenfurt Austria, May 2016, pp. 1–11.

[45] S. Lee and A. C. Bovik, “Fast algorithms for foveated video processing,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 2, pp. 149–162, February 2003.

[46] C. Guo and L. Zhang, “A novel multiresolution spatiotemporal saliency detection model and its applications in image and video compression, image processing,” *IEEE Trans. Image Process.*, vol. 19 Issue 1, pp. 185–198, 2010.

[47] H. Hadizadeh and I. V. Bajic, “Saliency-preserving video compression,” in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME ’11)*, 2011, pp. 1–6.

[48] L. Itti, “Automatic foveation for video compression using a neurobiological model of visual attention,” *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1304–1318, 2004.

[49] P. Silsbee, A. Bovik, and D. Chen, “Visual pattern image sequence coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 4, pp. 291–301, Aug 1993.

[50] T. H. Reeves and J. A. Robinson, “Adaptive foveation of MPEG video,” in *Proc. 4th ACM Int. Conf. on Multimedia*, ser. MULTIMEDIA ’96. New York, USA: ACM, 1996, pp. 231–241.

[51] S. Lee and A. Bovik, “Very low bit rate foveated video coding for H.263,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 6, Mar 1999, pp. 3113–3116.

[52] S. Lee, M. Pattichis, and A. Bovik, “Foveated video compression with optimal rate control,” *IEEE Trans. Image Process.*, vol. 10, no. 7, pp. 977–992, Jul 2001.

[53] R. S. Wallace, P. W. Ong, B. Bederson, and E. L. Schwartz, “Space variant image processing,” *Int. J. Computer Vision*, vol. 13, no. 1, pp. 71–90,, 1994.

[54] P. Kortum and W. S. Geisler, “Implementation of a foveated image coding system for image bandwidth reduction,” in *Proc. SPIE Human Vision and Electronic Imaging*, vol. 2657, San Jose, CA, April 1996, pp. 350–360.

[55] S. Lee and A. Bovik, “Foveated video image analysis and compression gain measurements,” in *Proc. 4th IEEE Southwest Symposium on Image Analysis and Interpretation*, Austin, Texas, April 2000, pp. 63–67.

[56] S. Azizi, D. Cochran, and J. N. McDonald, "A sampling approach to region-selective image compression," in *Proc. IEEE Conf. on Signals, Systems and Computers*, October 2000, pp. 1063–1067.

[57] M. Chessa and F. Solari, *Proc. 18th Int. Conf. on Image Analysis and Processing (ICIAP) , Part I*. Genoa, Italy,: Springer International Publishing, September 2015, ch. Local Feature Extraction in Log-Polar Images, pp. 410–420.

[58] W. S. Geisler and J. S. Perry, "A real-time foveated multiresolution system for low-bandwidth video communincation," in *Proc. SPIE*, vol. 3299, 1998.

[59] S. Lee, A. C. Bovik, and Y. Y. Kim, "Low delay foveated visual communications over wireless channels," in *Proc. IEEE Int. Conf. Image Process.*, Kobe, Japan, October 1999.

[60] Z. Wang, L. Lu, and A. C. Bovik, "Foveation scalable video coding with automatic fixation selection," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 243–254, 2003.

[61] X. Shang, Y. Wang, L. Luo, and Z. Zhang, "Perceptual multiview video coding based on foveated just noticeable distortion profile in DCT domain," in *Proc. 20th IEEE Int. Conf. on Image Processing (ICIP)*. IEEE, 2013, pp. 1914–1917.

[62] B. P. J. and A. E. H., "The laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, 1983.

[63] J. Hoffman, *Attention*. Hove, UK: Psychology Press, 1998, ch. Visual attention and eye movements, pp. 119–154.

[64] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

[65] R. Rensink, "Seeing, sensing, and scrutinizing," *Vision Research*, vol. 40, No. 10-12, pp. 1469–1487, June 2000.

[66] D. Walther and C. Koch, "Modeling attention to salient proto-objects," *Neural Networks*, vol. 19, no. 9, pp. 1395–407, 2006.

[67] Y.-F. Ma and H. Zhang, "Contrast-based image attention analysis by using fuzzy growing," in *Proc. 11th ACM Int. Conf. on Multimedia*, Berkeley, CA, USA, November 2003, pp. 374–381.

[68] F. Liu and M. Gleicher, "Region enhanced scale-invariant saliency detection," in *Proc. IEEE Int. Conf. on Multimedia and Expo*. Toronto, Ont.: IEEE, 2006, pp. 1477–1480.

[69] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency," in *Proc. 12th Annual Conf. Neural Information Processing Systems*, vol. 19. Vancouver, British Columbia, Canada: MIT Press, December 2006, pp. 545–552.

[70] D. Gao, V. Mahadevan, and N. Vasconcelos, "The discriminant center-surround hypothesis for bottom-up saliency," *Advances in Neural Information Processing Systems (NIPS)*, pp. 497–504, 2007.

[71] L. Itti and P. Baldi, "A principled approach to detecting surprising events in video," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2005)*, January 2005, pp. 631–637.

[72] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR 2007)*. Minneapolis, Minnesota, USA: Computer Society, June 2007.

[73] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2008)*. Anchorage, Alaska, USA: Computer Society, June 2008, pp. 1–8.

[74] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Ssstrunk, "Frequency-tuned salient region detection," in *Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR 2009),*, Miami, Florida, USA, June 2009, pp. 1597–1604.

[75] Y. Yu, B. Wang, and L. Zhang, "Pulse discrete cosine transform for saliency-based visual attention," in *Proc. Int. Conf. Development and Learning*. Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 1–6.

[76] X. Cui, Q. Liu, and D. Metaxas, "Temporal spectral residual: Fast motion saliency detection," in *Proc. 17th ACM Int. Conf. on Multimedia*, ser. MM '09. New York, USA: ACM, 2009, pp. 617–620.

[77] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," in *Proc. 23rd, IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2010)*, San Francisco, CA, USA, June 2010, pp. 2376–2383.

[78] V. Mahadevan and N. Vasconcelos, "Spatiotemporal saliency in dynamic scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 171–177, 2010.

[79] G. Hua, C. Zhang, Z. Liu, Z. Zhang, and Y. Shan, "Efficient scale-space spatiotemporal saliency tracking for distortion-free video retargeting," in *Computer Vision ACCV 2009*, ser. Lecture Notes in Computer Science, H. Zha, R.-i. Taniguchi, and S. Maybank, Eds. Springer Berlin Heidelberg, 2010, vol. 5995, pp. 182–192.

[80] Y. Fang, W. Lin, B.-S. Lee, C.-T. Lau, Z. Chen, and C.-W. Lin, "Bottom-up saliency detection model based on human visual sensitivity and amplitude spectrum," *IEEE Trans. Multimedia*, vol. 14, Issue 1, pp. 187–198, February 2012.

[81] N. Imamoglu, W. Lin, and Y. Fang, "A saliency detection model using low-level features based on wavelet transform," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 96–105, Jan 2013.

[82] S. Lu, C. Tan, and J. H. Lim, "Robust and efficient saliency modeling from image co-occurrence histograms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 195–201, Jan 2014.

[83] N. Imamoglu, E. Dorronzoro, M. Sekine, K. Kita, and W. Yu, "Spatial visual attention for novelty detection: A space-based saliency model in 3D using spatial memory," *IPSJ Transactions on Computer Vision and Applications*, vol. 7, pp. 35–40, April 2015.

[84] A. V. Oppenheim and J. S. Lim, "Importance of phase in signals," in *Proc. IEEE Conf.*, vol. 69, no. 5. IEEE, 1981, pp. 529–541.

[85] X. Hou, J. Harel, and C. Koch, "Image signature: Highlighting sparse salient regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, No. 1, pp. 194–201, January 2012.

[86] T. Lindeberg and B. M. ter Haar Romeny, *Geometry-Driven Diffusion in Computer Vision*, ser. Mathematical Imaging and Vision. Dordrecht, Netherlands: Kluwer Academic Publishers, 1994, vol. 1, ch. Linear scale-space, pp. 1–38.

[87] T. A. Ell and S. J. Sangwine, "Hypercomplex fourier transforms of color images," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 22–35, Jan. 2007.

[88] Y. Ying-li Tian and A. Hampapur, "Robust salient motion detection with complex background for real-time video surveillance," *IEEE Workshop on Motion and Video Computing*, vol. 2, pp. 30–35, 2005.

[89] R. Llados-Bernaus and R. Stevenson, "Fixed-length entropy coding for robust video compression," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 6, pp. 745–755, Oct 1998.

[90] H. Wang, N. man Cheung, and A. Ortega, "WZS: Wyner-Ziv scalable predictive video coding," in *Proc. Picture Coding Symposium (PCS)*, 2004, pp. 1–6.

[91] J. Ohm, M. van der Schaar, and J. W. Woods, "Interframe wavelet coding motion picture representation for universal scalability," *Signal Processing: Image Communication*, vol. 19, no. 9, pp. 877–908, October 2004.

[92] K. R. Rao and P. Yip, Eds., *The Transform and Data Compression Handbook*. Boca Raton: CRC Press LLC, 2001.

[93] F. A. Kamangar and K. R. Rao, "Fast algorithms for the 2-D discrete cosine transform," *IEEE Trans. Comput.*, vol. 31, no. 9, pp. 899–906, Sep. 1982.

[94] J. Liang and T. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3032–3044, Dec. 2001.

[95] ITU-T Recommendation T.81 / ISO/IEC 10918-1, *Digital compression and coding of continuous-tone still images*, ITU-T / ISO/IEC Std., 1992.

[96] G. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[97] F. Kamisli and J. S. Lim, "1-D transforms for the motion compensation residual," *IEEE Trans. Image Process.*, vol. 20, No. 4, pp. 1036–1046, April 2011.

[98] B. Zeng and J. Fu, "Directional discrete cosine transform - a new framework for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, No. 3, pp. 305–313, March 2008.

[99] R. A. Cohen, S. Klomp, A. Vetro, and H. Sun, "Direction-adaptive transform for coding prediction residuals," in *Proc. 17th IEEE Int. Conf. Image Process.*, Hong Kong, September 2010.

[100] C.-L. Chang, M. Makar, S. Tsai, and B. Girod, "Direction-adaptive partitioned block transform for color image coding," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1740–1755, July 2010.

[101] X. Peng, J. Xu, and F. Wu, "Directional filtering transform for image/intra-frame compression." *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2935–2946, 2010.

[102] Y. Wang, J. Chen, and Y. He, "A pixel-wise directional intra prediction method," *J. Vis. Commun. Image Representation*, vol. 23, no. 4, pp. 599 – 603, 2012.

[103] A. Gabriellini, M. Naccari, M. Mrak, D. Flynn, and G. V. Wallendael, "Adaptive transform skipping for improved coding of motion compensated residuals," *Signal Processing : Image Communication*, vol. 28, no. 3, pp. 197 – 208, 2013.

[104] M. Wang, K. N. Ngan, and L. Xu, "Efficient H.264/AVC video coding with adaptive transforms," *IEEE Trans. Multimedia*, vol. 15, No. 4, pp. 933–946, June 2014.

[105] C. Zhang, C. Wang, , and B. Jiang, "Color image compression based on directional all phase biorthogonal transform," *Int. J. Multimedia and Ubiquitous Engineering*, vol. 10, no. 1, pp. 247–254, January 2015.

[106] Y. Ye and M. Karczewicz, "Improved H.264 intra coding based on bi-directional intra predicton , directional transform, and adaptive coefficient scanning," in *Proc. of IEEE Int. Conf. on Image Processing*, October 2008, pp. 2116–2119.

[107] C.Yeo, Y.H.Tan, and Z. Li, "Low-complexity mode-dependent KLT for block-based intra coding," in *Proc. IEEE Int. Conf. on Image Processing (ICIP)*, Brussels, Belgium, Septermber 2011, pp. 3685–3688.

[108] C.Yeo, Y.H.Tan, Z.Li, and S. Rahardja, "Mode-dependent transforms for coding directional intra prediction residuals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, No. 4, pp. 545–554, April 2012.

[109] J. Han, A. Saxena, V. Melkote, and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1874–1884, April 2012.

[110] Z. Gu, W. Lin, B.-S. Lee, and C. T. Lau, "Rotated orthogonal transform (ROT) for motion-compensation residual coding," *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4770–4781, 2012.

[111] A. Saxena and F. C. Fernandes, "DCT/DST-based transform coding for intra prediction in image/video coding," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3974–3981, October 2013.

[112] X. Cai and J. S. Lim, "Algorithms for transform selection in multiple-transform video compression," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5395–5407, December 2013.

[113] S. Dikbas and Y. Altunbasak, "Novel true-motion estimation algorithm and its application to motion-compensated temporal frame interpolation," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 2931–2945, August 2013.

[114] M. V. Afonso, J. C. Nascimento, and J. S. Marques, "Automatic estimation of multiple motion fields from video sequences using a region matching based approach," *IEEE Trans. Multimedia*, vol. 16, no. 1, pp. 1–14, January 2014.

[115] A. Abou-elailah, F. Dufaux, J. Farah, M. Cagnazzo, and B. Pesquet-popescu, "Fusion of global and local motion estimation for distributed video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 158–172, January 2013.

[116] Z. Chen, P. Zhou, and Y. He, *Fast integer pel and fractional pel motion estimation for JVT*, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG Std. JVT-F017, December 2002.

[117] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Proc. SPIE Visual Communications and Image Processing*, C.-C. J. Kuo, Ed., vol. 4671, January 2002, pp. 1069–1079.

[118] V. Argyriou and T. Vlachos, "A study of sub-pixel motion estimation using phase correlation," in *Proc. Conf. British Machine Vision*. BMVA Press, 2006, pp. 40.1–40.10.

[119] I. Ishii, T. Taniguchi, K. Yamamoto, and T. Takaki, "High-frame-rate optical flow system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 1, pp. 105–112, January 2012.

[120] X. Chen, N. Canagarajah, J. L. Nunez-Yanez, and R. Vitulli, "Lossless video compression based on backward adaptive pixel-based fast motion estimation," *Signal Processing : Image Communication*, vol. 27, no. 9, pp. 961–972, 2012.

[121] M. Tok, A. Glantz, A. Krutz, and T. Sikora, "Monte-carlo-based parametric motion estimation using a hybrid model approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 4, pp. 607–620, April 2013.

[122] C. Dhoot, L. pui Chau, S. R. Chowdhury, and V. J. Mooney, "Low power motion estimation based on probabilistic computing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 1, pp. 1–14, January 2014.

[123] E. Monteiro, B. Vizzotto, C. Diniz, M. Maule, B. Zatt, and S. Bampi, "Parallelization of full search motion estimation algorithm for parallel and distributed platforms," *Int. J. Parallel Programming*, vol. 42, no. 2, pp. 239–264, April 2014.

[124] Y. Huang, B. Y. Hsieh, S. Chien, S. Ma, and L. Chen, "Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 507–522, April 2006.

[125] J. Jain and A. K. Jain, "Displacement measurement and its applications in interframe image coding," *IEEE Trans. Commun.*, vol. 29, pp. 1799–1808, 1981.

[126] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing," in *Proc. National Telecommunication Conf.*, Nov. 29-Dec. 3 1981, pp. G5.3.1–G5.3.5.

[127] R. Li, B. Zeng, and M. L. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438–42, August 1994.

[128] L.-M. Po and W.-C. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, June 1996.

[129] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim, "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 369–377, 1998.

[130] S. Zhu and K.-K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287–290, February 2000.

[131] C. Zhu, X. Lin, L.-P. Chau, K. P. Lim, H.-A. Ang, and C.-Y. Ong, "A novel hexagon-based search algorithm for fast block motion estimation," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2001, pp. 1593–1596.

[132] Z. Cui, D. Wang, G. Jiang, and C. Wu, "Octagonal search algorithm with early termination for fast motion estimation on H.264," in *IAS*. IEEE Computer Society, 2009, pp. 123–126.

[133] C.-H. Cheung and L.-M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168–1177, Dec 2002.

[134] 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.

[135] Y. Cheng, L. Yang, Z. Fang, H. Hou, and G. Chen, "A fast motion estimation algorithm based on diamond and hexagon search patterns," in *Proc. Joint Conferences on Pervasive Computing (JCPC)*, Dec 2009, pp. 595–598.

[136] H. Nisar, A. S. Malik, and T.-S. Choi, "Content adaptive fast motion estimation based on spatio-temporal homogeneity analysis and motion classification," *Pattern Recognition Letters*, vol. 33, no. 1, pp. 52–61, Jan. 2012.

[137] B. Abdoli, R. Sedaghat, and M. Taghiloo, "Optimized predictive zonal search (OPZS) for block-based motion estimation," *Signal Processing: Image Communication*, vol. 39, Part A, pp. 293–304, November 2015.

[138] A. M. Tourapis, A. Leontaris, K. S□hring, and G. Sullivan, *H.264/14496-10 AVC Reference Software Manual (JVT-AE010)*, 12th ed., Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG (ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6), 31st Meeting: London, UK, January 2009.

[139] K. M. Nam, J.-S. Kim, R.-H. Park, and Y. S. Shim, "A fast hierarchical motion vector estimation algorithm using mean pyramid," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 4, pp. 344–351, Aug 1995.

[140] Y.-C. Lin and S.-C. Tai, "Fast full-search block-matching algorithm for motion-compensated video compression," in *Proc. 13th Int. Conf. on Pattern Recognition*, vol. 3, Aug 1996, pp. 914–918.

[141] S.-J. Park, S.-M. Hong, H. Lee, S. Jin, and J. Jeong, "Histogram ordering model-based fast motion estimation," *Image Processing, IET*, vol. 6, no. 3, pp. 238–250, April 2012.

[142] C.-S. Park, "Multilevel motion estimation based on distortion measure in transform domain," *Electronics Letters*, vol. 49, no. 14, pp. 880–882, July 2013.

[143] A. Paramkusam and V. Reddy, "Two-layer motion estimation algorithm for video coding," *Electronics Letters*, vol. 50, no. 4, pp. 276–278, February 2014.

[144] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. on Neural Networks*, vol. 4, Perth, Australia, 1995, pp. 1942–1948.

[145] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. on Evolutionary Computation*. Washington, DC, USA: IEEE Computer Society, May 1998, pp. 69–73.

[146] ——, "Parameter selection in particle swarm optimization," in *Proc. 7th Int. Conf. on Evolutionary Programming VII*. London, UK: Springer-Verlag, 1998, pp. 591–600.

[147] T. Gong and R. Ding, "A modified genetic algorithm based block matching motion estimation method," *Signal processing*, vol. 19, no. 3, pp. 207–210, 2003.

[148] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, June 2004.

[149] R. Ren, M. Manokar, Y. Shi, and B. Zheng, "A fast block matching algorithm for video motion estimation based on particle swarm optimization and motion prejudgment," *ACM Symposium of Applied Computing(SAC)*, vol. abs/cs/0609131, no. cs.MM/0609131, pp. 1–10, Sep 2006.

[150] X. Yuan and X. Shen, "Block matching algorithm based on particle swarm optimization," in *Int. Conf. on Embedded Software and Systems (ICESS2008)*, Sichuan, China, 2008, pp. 191–195.

[151] E. Cuevas, "Block-matching algorithm based on harmony search optimization for motion estimation," *Applied Intelligence, Springer*, vol. 39, no. 1, pp. 165–183, December 2012.

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

[153] E. Cuevas, D. Zaldivar, M. Perez-Cisneros, H. Sossa, and V. Osuna, "Block matching algorithm for motion estimation based on artificial bee colony (ABC)," *Applied Soft Computing, Elsevier*, vol. 13, no. 6, pp. 3047–3059, June 2013.

[154] S. I. A. Pandian, G. J. Bala, and J. Anitha, "A pattern based pso approach for block matching in motion estimation," *Engineering Applications of Artificial Intelligence, Elsevier*, vol. 26, no. 8, pp. 1811–1817, September 2013.

[155] C. Fei, P. Zhang, and J. Li, "Motion estimation based on artificial fish-swarm in H.264/AVC coding," *WSEAS Transactions on signal processing*, vol. 10, pp. 221–229, 2014.

[156] M. K. Jalloul and M. A. Al-Alaoui, "A novel cooperative motion estimation algorithm based on particle swarm optimization and its multicore implementation," *Signal Processing: Image Communication*, vol. 39, Part A, pp. 121–140, November 2015.

[157] B. A. Wandell, *Foundations of Vision*. Sinauer Associates, Sunderland, Massachusetts, 1995.

[158] S. Shimizu, "Wide-angle foveation for all-purpose use," *IEEE-ASME Trans. Mechatronics*, vol. 13, no. 5, pp. 587–597, Oct 2008.

[159] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *Proc. 24th IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, Colorado Springs, CO, USA, June 2011, pp. 409–416.

[160] R. Achanta and S. S□sstrunk, "Saliency detection for content-aware image resizing," in *Proc. of the Int. Conf. on Image Process. (ICIP)*. Cairo, Egypt: IEEE, November 2009, pp. 1005–1008.

[161] A. Koz and A. Aydin Alatan, "Foveated image watermarking," in *Proc. Int. Conf. on Image Processing*, vol. 3, June 2002, pp. 657–660.

[162] Z. Ma, L. Qing, J. Miao, and X. Chen, "Advertisement evaluation using visual saliency based on foveated image," in *Proc. IEEE Int. Conf. Multimedia and Expo (ICME 2009)*. New York City, USA: IEEE, June 2009, pp. 914–917.

[163] R. B. Gomes, B. M. F. da Silva, L. K. de Medeiros Rocha, R. V. Aroca, L. C. P. R. Velho, and L. M. G. Goncalves, "Efficient 3D object recognition using foveated point clouds," *Computers & Graphics*, vol. 37, no. 5, pp. 496 – 508, 2013.

[164] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," *Human Neurobiology*, vol. 4, no. 4, pp. 219–227, 1985.

[165] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *Journal of Physiology*, vol. 160, pp. 106–154, 1962.

[166] C. Ware, *Visual thinking for design*. Morgan Kaufmann, 2010.

[167] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden, "Pyramid methods in image processing," *RCA Engineer*, vol. 29, no. 6, pp. 33–41, 1984.

[168] W. K. Pratt, *Digital Image Processing: PIKS Inside*, 3rd ed. New York, USA: John Wiley & Sons Inc., 2001.

[169] A. Van Der Schaaf and J. H. Van Hateren, "Modeling the power spectra of natural images: Statistics and information," *Vision Research*, vol. 36, pp. 2759–2770, 1996.

[170] T. I. Haweel, "A new square wave transform based on the DCT," *Signal Processing*, vol. 81, no. 11, pp. 2309 –2319, November 2001.

[171] Z. Li, "A saliency map in primary visual cortex," *Trends in Cognitive Sciences*, vol. 6, No. 1, pp. 9–16, 2002.

[172] L. Zhaoping, *Neurobiology of attention*. Elsevier, 2005, ch. The primary visual cortex creates a bottom-up saliency map, pp. 570–575.

[173] D. Mahapatra, S. Winkler, and S.-C. Yen, "Motion saliency outweighs other low-level features while watching videos," *SPIE- Human Vision and Electronic Imaging XIII*, vol. 6806, pp. 27–31, Jan 2008.

[174] (2007) MSRA salient object database. [Online]. Available: http://research.microsoft.com/en-us/um/people/jiansun/SalientObject/salient_object.htm

[175] R. Achanta. (2009) Ground truth of 1000 images. [Online]. Available: http://ivrgwww.epfl.ch/supplementary_material/RK_CVPR09/GroundTruth/binarymasks.zip

[176] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[177] W. Mauersberger, "Generalised correlation model for designing 2- dimensional image coders," *Electronic Letters*, vol. 15, no. 20, p. 664□665, September 1979.

[178] A. K. Jain, "A sinusoidal family of unitary transforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, NO. 4,, p. 356□365, 1979.

[179] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan 1974.

[180] E.Alshina, A.Alshin, and F.C.Fernandes, "Rotational transform for image and video compression," in *Proc. IEEE Int. Conf. on Image Process. (ICIP)*, Brussels, BE, September 2011.

[181] P. Kauff and K. Schuur, "Shape-adaptive DCT with block-based DC separation and ΔDC correction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 237–242, Jun 1998.

[182] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, July 2003.

[183] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[184] M. R. Pickering, J. F. Arnold, and M. R. Frater, "An adaptive search length algorithm for block matching motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 6, pp. 906–912, December 1997.

[185] M. Accame, F. G. B. D. Natale, and D. D. Giusto, "Hierarchical motion estimator (HME) for block-based video coders," *IEEE Trans. Consumer Electronics*, vol. 43, no. 4, pp. 1320–1330, November 1997.

[186] J.-J. Tsai and H.-M. Hang, "Modeling of pattern-based block motion estimation and its application," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 1, pp. 108–113, Jan 2009.

[187] C.-W. Lam, L.-M. Po, and C. H. Cheung, "A novel kite-cross-diamond search algorithm for fast block matching motion estimation," in *Proc. Int. Symposium on Circuits and Systems (ISCAS '04)*, vol. 3, May 2004, pp. III–729–32.

[188] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization : an overview," *Swarm Intelligence Journal*, vol. 1, no. 1, pp. 33–57, August 2006.

[189] R. Eberhart and S. Y. Shi, "Particle swarm optimization: developments, applications and resources," in *Proc. Congress on Evolutionary Computation*, vol. 1, 2001, pp. 81–86.

[190] P. Angeline, "Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences," in *Evolutionary Programming VII*, ser. Lecture Notes in Computer Science, V. Porto, N. Saravanan, D. Waagen, and A. Eiben, Eds. Springer Berlin Heidelberg, 1998, vol. 1447, pp. 601–610.

[191] Y. Shi and R. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congress on Evolutionary Computation (CEC 99)*, vol. 3, 1999, pp. 1945–1950.

[192] P. K. Tripathi, S. Bandyopadhyay, and S. K. Pal, "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients," *Information Sciences*, vol. 177, no. 22, pp. 5033 – 5049, 2007.

[193] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," in *Proc. Congress on Evolutionary Computation (CEC 99)*, vol. 3, 1999, pp. 1951–1957.

[194] A. Silva, A. Neves, and E. Costa, "An empirical comparison of particle swarm and predator prey optimisation," in *Proc. 3th Irish Int. Conf. on Artificial Intelligence and Cognitive Science*, ser. Lecture Notes in Computer Science, vol. 2464. Springer Berlin Heidelberg, 2002, pp. 103–110.

[195] M. Pant, T. Radha, and V. Singh, "A simple diversity guided particle swarm optimization," in *Proc. IEEE Congress on Evolutionary Computation (CEC 2007)*, Sept 2007, pp. 3294–3299.

[196] Y. Liu, Z. Qin, Z.-L. Xu, and X. shi He, "Using relaxation velocity update strategy to improve particle swarm optimization," in *Proc. Int. Conf. on Machine Learning and Cybernetics*, vol. 4, Aug 2004, pp. 2469–2472.

[197] Y.-C. Lin and S.-C. Tai, "Fast full-search block-matching algorithm for motion-compensated video compression," *IEEE Trans. Commun.*, vol. 45, no. 5, pp. 527–531, May 1997.

[198] X. Yi, J. Zhang, N. Ling, , and W. Shang, *Improved and simplified fast motion estimation for JM*, Joint Video Team of ISO/IEC MPEG & ITU-T VCEG Std., July 2005.

Dissemination of Research Outcome

Journals

1. **Deepak Singh** and Sukadev Meher, "An Efficient Multiscale Phase Spectrum based Salient Object Detection Technique," *International Journal of Computer Application (IJCA)*, Vol 3, pp. 29-33, February 2013.

2. **Deepak Singh** and Sukadev Meher, "Direction-Adaptive Motion Estimation (DAME) for Efficient Video Compression," *IETE Technical Review*, Taylor & Francis., Vol 33, Issue 3, pp. 231-243, 2016.

3. **Deepak Singh** and Sukadev Meher, "Direction-Adaptive Fixed Length Discrete Cosine Transform Framework for Efficient H.264/AVC Video Coding," *Journal of Signal Processing Image Communication*, Elsevier, Vol 48, pp. 22-37, Oct 2016.

Conferences

1. **Deepak Singh** and Sukadev Meher, "A novel approach for saliency detection based on multiscale phase spectrum," *IEEE Int. Conf. on Communication, Information & Computing Technology (ICCICT)*, Mumbai, pp. 1-6, 2012.

2. **Deepak Singh** and Sukadev Meher, "An Efficient Multiscale Phase Spectrum based Salient Object Detection Technique," *Int. Conf. on Electronic Design and Signal Processing ICEDSP(3)*, Manipal, pp. 29-33, 20-22, December 2012.