CARNEGIE MELLON UNIVERSITY

STEREOSCOPIC IMAGE SEQUENCE COMPRESSION USING MULTIRESOLUTION AND QUADTREE DECOMPOSITION BASED DISPARITY- AND MOTION-ADAPTIVE SEGMENTATION

A DISSERTATION
SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree of

DOCTOR OF PHILOSOPHY
in
ELECTRICAL ENGINEERING

by

SRIRAM SETHURAMAN

Pittsburgh, Pennsylvania
July, 1996

Abstract

Stereoscopic image display offers a simple and compact means of portraying on 2D screens the relative depth information in a real world scene. In addition to serving as a disambiguating cue, the perception of depth considerably enhances the viewing experience. Typically, more than two views would have to be transmitted either to provide the correct perspective to each viewer in a multi-viewer scenario or to provide a single viewer with the feel of “look-around”. This results in a multi-fold increase in bandwidth over the existing monoscopic channel bandwidths. Achieving a significant reduction in the excess bandwidth needed for coding stereoscopic video, over the bandwidth required for independent coding of these multiple views, is the primary objective of this thesis. To this end, we present a framework for stereoscopic image sequence compression that brings together several computationally efficient algorithms in a unified fashion to address critical issues such as, (1) tailoring the excess bandwidth to be commensurate with the demand for stereoscopic video, (2) compatible coding (in terms of quality and technology), (3) scalability of coding efficiency and computational complexity with multiple views, and (4) synthesis of intermediate views to provide motion parallax perception to the viewer.

At the heart of the framework is a computationally efficient multiresolution and quadtree decomposition based segmentation scheme that provides an optimal representation for coding a stereoscopic image sequence by jointly minimizing the overhead for coding the segmentation structure and the segment disparities or displacements. The framework exploits both intra-view and inter-view redundancies to predict temporal and perspective-induced occlusions. Two configurations for stereoscopic sequence coding are investigated. In the first configuration, a complete disparity map is available at the decoder for each stereo-frame; this disparity map can be used to synthesize intermediate views. In the second configuration, the coding efficiency is improved by relaxing this restriction. We also present two segment-tracking-based joint coding approaches, whose coding efficiency and complexity scale well with multiple views. A scan-line method that exploits the estimated disparity map and the imaging geometry is used to fill-in regions that are uncovered during segment tracking.

For the sake of compatibility with monoscopic transmission, we code one view of the multi-view sequence at a higher quality than the other views. A quadtree-based residual coding method that is suited for low bit-rate coding is used to achieve a graceful degradation of quality at low bit-rates. Also, our framework supports the psychophysically-motivated mixed-resolution-based coding to achieve a further reduction in the excess bandwidth without affecting the perceived stereoscopic quality. Over a test set of stereoscopic sequences the performances of our stereoscopic sequence coding extensions are presented and compared with the performance of two baseline schemes, one for each configuration, that are representative of the approach used by current international stereoscopic video coding standards, e.g., MPEG-2 standards.

Acknowledgements

I would like to express my sincere gratitude to my advisors, Prof. Angel G. Jordan and Prof. Mel Siegel, for their constant encouragement, advice, support, and patience (!). I thank the other members of my thesis committee, Prof. Moura and Prof. Witkin, for their time, effort and suggestions. I would like to thank fellow graduate students in the research group, Tom Ault, Priyan Gunatilake, Jeff McVeigh, and Huadong Wu, with whom I have had several useful discussions. My thanks are also due to other members of the research group, Alan Guisewite, Victor Grinberg, Gregg Podnar, and Scott Safier, for their kind assistance and camaraderie. I thank Dr. James Tam of the Communication Research Center, Ottawa for providing us with the tape containing the DISTIMA test sequences. I would also like to thank CCETT, France for making these test sequences available for research purposes.

My sincere thanks to the secretaries at CIMDS, Carol Boshears and Michelle Agie, for their help. I would also like to thank Elaine Lawrence and Lynn Philibin of the ECE graduate office for their cordial and friendly help with the department matters.

I am eternally indebted to my family members for their love, support and advice which words cannot describe.

Last but not the least, I would like to thank my other friends and fellow Indian classical music lovers of Pittsburgh who have made my stay in Pittsburgh memorable and enjoyable.

This research was supported by the Advanced Research Projects Agency under ARPA Grant No. MDA 972-92-J-1010.

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations
(first occurrence)

BM - Block matching ..... (16)
bpp - bits per pixel ..... (21)
CIF - Common intermediate format ..... (19)
CST - Contrast sensitivity threshold ..... (13)
DBS - Disparity based segmentation ..... (5)
DBS-1 - Stereoscopic sequence coding extension using DBS - configuration 1 ..... (64)
DBS-2 - Stereoscopic sequence coding extension using DBS - configuration 2 ..... (64)
DC - Disparity compensation ..... (3)
DCP - Disparity compensated prediction ..... (29)
DCT - Discrete cosine transform ..... (13)
DFD - Displaced frame difference ..... (16)
DPCM - Differential pulse code modulation ..... (12)
EPI - Epipolar plane image ..... (88)
FBS - Fixed block-size ..... (4)
FBS-BMA - Fixed block-size based block matching algorithm ..... (30)
FBS-1 - Stereoscopic sequence coding extension using FBS-BMA - configuration 1 ..... (63)
FBS-2 - Stereoscopic sequence coding extension using FBS-BMA - configuration 2 ..... (63)
HBM - Hierarchical block matching ..... (17)
HDTV - High definition television ..... (19)
HMD - Head mounted display ..... (9)
HVS - Human Visual system ..... (12)
IVS - Intermediate view synthesis ..... (3)
LBG - Linde, Buzo, Gray algorithm ..... (14)
LOT - Lapped orthogonal transform ..... (14)
MR - Multiresolution ..... (5)
MSE - Mean squared error ..... (14)
MAD - Minimum absolute difference ..... (16)
MB - Macroblock ..... (18)
MCP - Motion compensated prediction ..... (16)
MDBS - Motion and disparity based segmentation ..... (68)
ME - Motion estimation ..... (16)
MF - Model failure ..... (18)
MPEG - Motion picture experts group ..... (19)
MRE - Multiresolution estimation ..... (33)

MRFB - Multirate filter bank ..... (23)
MR-QTD - Multiresolution based quadtree decomposition ..... (56)
OBMC - Overlapped block motion compensation ..... (19)
PSNR - Peak signal-to-noise-ratio ..... (21)
QMF - Quadrature mirror filter ..... (24)
QTD - Quadtree decomposition ..... (33)
RDBS - Stereoscopic sequence coding extension using reversed DBS - configuration-1 (68)
SIF - Source input format ..... (19)
SSC - Stereoscopic sequence compression ..... (56)
SQ - Scalar quantizer ..... (13)
ST-1 - Stereoscopic sequence coding extension - Segment tracking - configuration-1 . (69)
VBS - Variable block-size (VBS) ..... (15)
VQ - Vector quantization ..... (12)
VLC - Variable length code ..... (63)

Chapter 1

Introduction

In this thesis, we present an efficient framework for coding stereoscopic video sequences that brings together several aspects of the problem in a coherent and unified manner. In Section 1.1, we present a broad picture of the status of stereoscopic imaging, display, and transmission today, and introduce the need for research in the area of stereoscopic sequence compression. We outline, in Section 1.2, some key contributions to this area and build up the motivation behind this thesis. In Section 1.3, we define the objectives of this thesis and briefly discuss the approach that we take to achieve them in Section 1.4. In Section 1.5, we summarize the major contributions of this thesis. The overview of the rest of the thesis is presented in Section 1.6.

1.1 PROBLEM BACKGROUND

Stereoscopic image display provides a simple means of perceiving the relative depth information in a real world scene from two suitably obtained images of that scene. The relative depth information can be vital in applications such as aerial photography (for 3D relief mapping), telemanipulation, surgery and microscopy. On the entertainment side, such as in televisions and video games, the added realism arising from the ability to perceive relative depth significantly enhances the viewing experience.

The concepts behind stereoscopic imaging and display have been in vogue for more than a century now. When compared to still stereo-photography, stereoscopic motion pictures and video did not gain widespread user acceptance owing to technological and psychological reasons. Some of the impediments to their popularity were, flicker, low spatial resolution, cross-talk between the left and right eye views, incorrect perspective, unnatural scene rotation with head movements, the need for a special eyeware, incorrect imaging geometries and increased eye-strain during long-term viewing. The recent advances in the imaging and display fields have resulted in increased spatial resolution, reduced flicker and minimal cross-talk. The new cost-effective autostereoscopic displays offer multiple viewing zones thus providing the correct perspective for each viewer in a multi-viewer scenario, and also dispense with the requirement of special eyeware (considered to be a serious factor in user acceptance). Television (TV) has gained widespread acceptance over the years and digital TVs are set to replace the analog TVs. The increased computing power that will

be present in the new generation of digital TVs and the proposed high-definition TVs can be exploited to provide viewers with geometrically correct perspectives based on head or eye tracking. This can significantly reduce the unnatural scene rotation with head movements and would also provide the viewer with motion parallax or the ‘look around’ feeling. The computing power can also be employed to adjust the screen disparity to suit the viewer’s taste, thus eliminating unnecessary eye-strain. Thus, with most of the serious handicaps being overcome, stereoscopic television (or 3DTV as it is more popularly known) is poised to become more acceptable among viewers.

However, the major problem associated with stereoscopic video transmission is the two-fold increase in the bandwidth needed to transmit the two views when compared to the transmission of a single view, if each of the views is coded independent of the other. In addition, a conventional stereoscopic image pair provides only the 3D view of a scene from a particular pair of viewpoints. To provide the ‘look-around’ effect discussed above or to accommodate multiple viewers, multiple viewing zones and correspondingly multi-view images are needed. This would result in an multi-fold increase in the transmission bandwidth. With the ever increasing demand for bandwidth, such an increase is neither acceptable nor practical. To tackle the bandwidth explosion associated with digital transmission, existing single view sequence transmission schemes already achieve a high degree of compression by exploiting spatial and temporal redundancies, and human visual system properties[1]. Hence, achieving any further significant reduction in bandwidth compared to independent coding of the sequences, requires a careful consideration of additional properties that can be exploited within the multi-view sequence, and other possible trade-offs between quality and bandwidth.

As the multi-view sequences contain essentially the same scene content taken from slightly different points of view, there exists a high degree of correlation between the sequences. It is possible to reduce the n-fold bandwidth requirement by exploiting this cross-stream correlation. Further reduction can be achieved by exploiting the geometric constraints imposed by the imaging setup (e.g., parallel camera axes), by employing object-oriented coding techniques, and by taking advantage of psychophysical masking effects present in human stereoscopic perception.

1.2 PRIOR WORK AND MOTIVATION

The use of stereo images for extracting depth information is a common technique in the field of machine vision and several intensity-based and feature-based image registration methods have evolved since the mid 70’s to solve the correspondence problem. However, typically, these methods are computationally intensive and are not directly suited for coding applications. The

1. Analog transmission methods also exploit some of these factors to achieve bandwidth reduction.

paper by Lukacs [86] constitutes one of the earliest attempts at multi-view coding in which the concept of disparity compensation (DC) (- establishing correspondence between similar areas in two images using the binocular disparity relationship - discussed in detail in Chapter 3) is used to predict the rest of the views from an independently coded view. Several DC-based methods have followed. Perkins [30] formalized DC-based coding as a conditional coding approach that is optimal for lossless coding and sub-optimal for lossy coding. He was also one of the first advocates of *mixed resolution coding*, wherein one of the views is presented at a lower resolution. A similar solution was also suggested by Dinstein [89]. Tamtaoui and Labit [31, 88] presented a constrained motion and disparity estimation scheme based on a calibrated pair of converged cameras. They also provided a coherence equation to verify the motion and disparity components between two stereo pairs of frames. Liu and Skerjanc [33] presented a dynamic programming based disparity estimation approach for establishing correspondence between edges. Tzovaras *et al*. [34] provided a hierarchical block matching method for disparity estimation. They were also the first to propose a bidirectional motion/disparity compensation which they called fused estimation. A genetic algorithm based disparity estimation solution was presented by Franich [91]. This work also introduced a smoothness measure to evaluate the disparity map. Ziegler and Panis [100] describe an object oriented coder for stereoscopic video coding along the lines of Hotter's [56] analysis-synthesis based coder. However, this method is useful only when there is no camera motion and when there are not many objects in the scene. Tzovaras *et al.* [99] have also recently proposed an object oriented coder that extracts the 3D motion model of objects based on the depth computed from the disparity map, and segments the frames based on depth and motion. However, no conclusive results are shown to indicate general applicability of such a coder. Puri *et al*. [96] present results of MPEG-2 compatible coding. One of the views is coded in the base layer and the other view is coded within the enhancement layer of the temporal scalability model of the MPEG-2 standard. A perceptually adaptive quantization scheme for such a compatible coder is presented in [101]. Some new algorithms have been reported in the area of multi-view coding and intermediate view synthesis (IVS) as well. Fujii *et al.* [93] describe a disparity estimation based multi-view coding and interpolation scheme. The encoding procedure extracts the 3D structure and texture of the scene. The disparity is computed on triangular patches using an affine model. The encoding procedure is computationally intensive even for simple scenes. However, good compression and interpolation results on two simple scenes are reported. A three camera based view interpolation using dynamic programming based disparity estimation is presented in [32]. Occlusions are minimized in this case by the triangular camera setup with two horizontally separated cameras and one vertically offset camera in between and above the other two cameras.

Thus, a lot of prior work has been done on the problems of stereoscopic image pair coding, mixed resolution stereo coding, joint estimation of motion and disparity, object oriented stereo coding, standards compatible stereo coding, psychophysically based bit allocation, multiresolution based stereo coding, multi-view coding and intermediate view synthesis. The DC-based

algorithms that have their origins in the computer vision field [33, 31] are primarily targeted at scene analysis and understanding and hence are computationally intensive and do not lend themselves well to coding. Advanced model-based coding methods [40, 100, 99], which use object-oriented 3D models, do not scale well with multiple objects in the scene; they are also computationally more complex than what the current technology can support in real-time. Other DC-based algorithms in the literature use simple fixed block-size (FBS) based disparity estimation [30, 87, 96, 34], which can be supported by the current technology. However, the bit allocation in these schemes is not commensurate with the local disparity or motion detail present within a multi-view sequence. These methods also suffer from spurious matches that affect the smoothness of the estimated disparity or motion field. A smooth disparity field is critical for intermediate view synthesis. Most of the schemes code all the views at the same quality and thus do not scale well with multiple views. Some other schemes code one sequence at a good quality and arbitrarily fix the bit-rate for the other streams. Satisfying such arbitrary bit-rate constraints can result in inadequate residual coding. This can give rise to objectionable artifacts that can cause confusion to the viewer when the resulting sequences are viewed stereoscopically. Except for [34, 96], most other methods consider only disparity compensation for coding multi-view sequences. This results in poor performance when a still image pair compression problem is extended to a sequence compression problem. Also, perspective-induced occlusions are not coded efficiently by these methods. While most of the coding advantages in video compression have been achieved by exploiting the tolerances of the human visual system, only few researchers [30, 87, 101] have exploited special stereoscopic masking effects to achieve large stereoscopic compression gains.

Hence the motivation behind the current work is to present a unified framework for stereoscopic image sequence compression that overcomes as many of these drawbacks as possible, and, at the same time, draws upon the desirable features of the different methods presented above. The specific objectives of this thesis are outlined in the next section.

1.3 OBJECTIVES

Though stereoscopic video provides added realism, it is our conjecture that the overall demand for stereo, over time and across viewers in a broadcast-type application, may never be high enough to warrant coding all views of a multi-view sequences at a high quality and resolution. Since most viewers are likely to be viewing monocularly at any given time, one of the views has to be coded at a higher visual quality. Hence, the foremost objective of this thesis is to develop a framework for stereoscopic image sequence compression within which quality compatibility with monoscopic transmission is maintained, and the *excess bandwidth* needed to code the rest of the views is commensurate with the perceived demand and functional advantages of stereoscopic video. The view that is coded at a high visual quality will hereafter be referred to as the *main sequence* and the other sequences will be referred to as *auxiliary sequences*. The

performance of the compression scheme should scale well with multiple views and should be able to take advantage of intra-view and inter-view redundancies. The computational complexity and performance of the coding scheme should be fairly independent of the nature of the scene in order to be applicable to any general scene. To be realizable in real-time using current technology, the proposed coding scheme should have a moderate encoder complexity and low decoder complexity. The coding scheme should be able to provide a smooth, disparity discontinuity preserving disparity map for intermediate view synthesis at the decoder. As mixed resolution coding can potentially help achieve low excess bandwidths, the coding scheme should be able to incorporate it in a straightforward fashion. The coding scheme should also offer standard sequence coding features such as, random-access, editability and independent decoding of sequence segments.

1.4 APPROACH

In this section, the approach that is taken to achieve the objectives set forth in Section 1.3 are outlined. Without loss of generality, only two views are considered in this section. Where necessary, information pertaining to more than two views is provided.

We initially consider an efficient method for coding stereoscopic still-images using disparity compensation. In contrast to high bit-rate high quality image coding, where the bits needed for coding the residuals in under-compensated regions typically constitute a major portion of the total bit-budget, the bits needed to code the disparities become significant at low bit-rates. Hence we need a scheme that can minimize the disparity coding bits for a given stereoscopic image pair. Also, to achieve low bit rates for the auxiliary stream, it becomes necessary to improve the compensation achieved through disparity estimation, so that the bit allocation for residual coding can be reduced. Better compensation as well as low disparity coding overhead can be achieved by resorting to content-adaptive disparity estimation. This leads us to the formulation of a *disparity-based segmentation* (DBS) scheme, which adapts the segment size to the local binocular disparity detail present in a stereoscopic image pair and thus minimizes the number of segment disparities that need to be coded. Moderate encoder complexity is maintained by incorporating the segmentation within a multiresolution (MR) framework. The MR framework aids in reducing the segmentation coding overhead and also automatically supports mixed resolution coding. The DBS scheme provides a smooth, accurate, and disparity-discontinuity-preserving disparity map that is well suited for intermediate view synthesis. The DBS algorithm is extended to code multiple views as well. The disparity map with multiple views is made more accurate by considering a multiple-baseline disparity estimation method.

Once a good disparity compensation is achieved, the areas with correspondence in the two views are well taken care of. But, those regions in the image being estimated which are occluded in the other view will remain uncompensated. When the stereoscopic still-image pair compression problem is extended to a stereoscopic sequence compression problem, these regions can be

estimated from a past or future frame within the auxiliary sequence (provided that these regions do not suffer from motion occlusions as well). Our framework takes advantage of this to further minimize the amount of residual coding needed. Since segmentation requires additional computations and a coding overhead to represent the location and size of the segments, several DBS-based coding extensions which try to minimize either the computational overhead, the coding overhead or both, are considered. The DBS algorithm provides a physical attribute to each segment, namely its disparity (or depth), which is not available when only one camera is used. Also, all pixels within a segment are likely to undergo similar displacement over time. Thus, by properly utilizing the segmentation and the segment disparity information, the main stream can also be compressed efficiently. By resorting to such joint coding, our framework aims to achieve lower bit-rates on the main stream as well.

1.5 IMPORTANT CONTRIBUTIONS OF THE THESIS

Some of the key contributions of this thesis are:

A novel multiresolution and quadtree decomposition based binocular disparity-adaptive segmentation of a stereoscopic image pair that is computationally efficient and results in a 25-55% saving in bits over fixed block-size based disparity compensation methods. (The segmentation technique can also be used with other homogeneity criteria to speed up the computations and to reduce segmentation coding overhead.)

A computational procedure for optimizing the rate-distortion performance of the DBS algorithm.

A motion- and disparity-adaptive extension of the above segmentation scheme that exploits intra-view and inter-view correlations to achieve efficient compression of stereoscopic image sequences by handling motion-based and perspective-induced occlusions properly.

A scheme for handling regions that are uncovered while tracking segments from one view to another, based on the imaging geometry and the estimated disparity map.

A computationally efficient segment tracking scheme that greatly simplifies motion and disparity estimation (by applying spatio-temporal coherence relationships), and that has a compression efficiency that scales well with multiple views.

A quadtree and vector quantization based residual coding scheme that provides better control over bit allocation than DCT-based residual coding.

An efficient approach to mixed resolution coding using a multiresolution framework.

Improved intermediate view synthesis through DBS and multiple-baseline disparity estimation.

1.6 THESIS OVERVIEW

The thesis chapters are structured as follows:

In Chapter 2 we provide background information on stereoscopy, digital video compression and multiresolution-based video coding. Concepts from these three areas serve as the foundations on which the later chapters are built. Chapter 2 also presents state-of-the-art information in each of these fields to aid the reader to better understand the underpinnings behind this work.

In Chapter 3 we introduce the problem of stereoscopic image compression, discuss prior work, reiterate the objectives and the need for a new framework, and present the disparity-based segmentation scheme. We discuss the components of the algorithm individually before outlining the entire algorithm. We also outline a computational method for optimizing the rate-distortion performance of the algorithm. We present coding results on several stereoscopic image pairs and compare them with the results obtained from FBS-based disparity compensation. Finally, we outline the computational efficiency of the DBS algorithm by comparing its complexity with that of a FBS-based scheme.

In Chapter 4 we extend the DBS algorithm to fit within a sequence coding framework. We present two major configurations in which this extension can be classified. Two baseline FBS-based schemes in each of these configurations, against which the performance of the DBS-based extensions can be compared, are developed. Several variants of sequence compression schemes incorporating the DBS, including two joint coding schemes, are presented within the framework. The rate-distortion performance of each variant over a test set of stereoscopic sequences is presented and compared against the baseline schemes. We also describe the psychophysical basis for mixed resolution coding and evaluate the compression gains that it offers

In Chapter 5 multi-view extensions of the DBS-based schemes are presented. A multiple-baseline DBS scheme that improves the accuracy of the disparity map in featureless areas is outlined. Intermediate views synthesized using the improved disparity map obtained from such a segmentation are compared with views synthesized using the disparity map obtained from a FBS-based disparity estimation scheme.

In Chapter 6 we summarize the results from Chapters 3, 4 and 5, emphasize the important contributions of the thesis, and present future directions.

In Appendix A, sample images from the stereoscopic image pairs and sequences used in this thesis, the camera geometry used to obtain the images (if available) and other details such as, frame rates, image sizes, maximum frame-to-frame motion and maximum disparity between views are included.

Chapter 2

Background

In this chapter, we present some necessary background information on which the rest of the chapters are based. In Section 2.1, fundamentals of stereoscopy are presented with subsections on human stereopsis, stereoscopic imaging, stereoscopic displays and imaging geometries. In Section 2.2, a reasonably detailed introduction to image and video compression is presented. Two decades of work and standardization activities in this field are summarized to provide a broader picture. In Section 2.3, the multiresolution framework that is extensively used in the later chapters is introduced and the advantages of the framework are stressed.

2.1 FUNDAMENTALS OF STEREOSCOPY

2.1.1 Stereopsis

Binocular stereopsis is the process primarily responsible for the perception of relative depth. The relative depth information in the scene being viewed is deduced by the human brain, based on the horizontal shift between corresponding points in the two-dimensional (2D) projections of a three-dimensional (3D) scene on the left and right retinae. The shift is known as the *retinal disparity* or *binocular parallax* of the corresponding real world point. Stereopsis can be used to 'trick' the brain into perceiving the relative depth in a 3D scene by presenting 2D projections of that scene, generated in such a way that the retinal projections of these 2-D projections are indistinguishable to the respective eyes from (or very similar to) the direct retinal projections of the 3-D scene.

2.1.2 Stereoscopic imaging

The process of acquiring two such 2D projections is known as *stereoscopic imaging*; it serves as a simple and compact means for capturing the relative depth information in a 3D scene as seen from a particular pair of viewpoints. Stereoscopic imaging imitates the human optical system in the sense that the two 2D projections, called a *stereoscopic image pair*, are obtained by imaging the 3D scene onto two appropriately positioned imaging sensors (photofilms or CCDs) through two imaging elements (lenses) that are separated by a distance called the *camera baseline*. The appropriate camera baseline, when the stereoscopic image pairs are intended for viewing, is the nominal interocular separation in humans. Stereoscopic imaging is also used in computer vision

systems to automatically compute the depth information in a scene from a stereoscopic image pair, based on the knowledge of the imaging geometry and the disparity between corresponding points in the left and right views. However, these systems often use a wider camera baseline to improve the accuracy of the depth estimates.

2.1.3 Stereoscopic displays

The recorded stereoscopic image pairs are to be suitably projected back on the left and right display screens (which can be a single physical display screen also) of a *stereoscopic display* for stereoscopic viewing. A suitable mechanism is needed to make each eye see only the image intended for it. The distance between corresponding points on the two display screens, when the two display screens are superimposed, is called the *screen disparity* of the corresponding real world point. The more common stereoscopic displays, head mounted displays (HMDs) excepted, use the same screen to display both images. Space, angle, time, color and polarization multiplexing schemes are used to multiplex the two eye views that are projected on the same screen. Due to engineering limitations, such multiplexing may require compromising, for each eye, one or more of spatial resolution, refresh rate, color components, and perceived brightness.

2.1.4 Stereoscopic imaging geometry

The relative positions and orientations of the two imaging elements and the two imaging sensor planes in a stereoscopic imaging setup constitute the *stereoscopic imaging geometry.* A stereoscopic imaging setup is shown in Fig. 2.1. A point P in the 3D scene is projected in

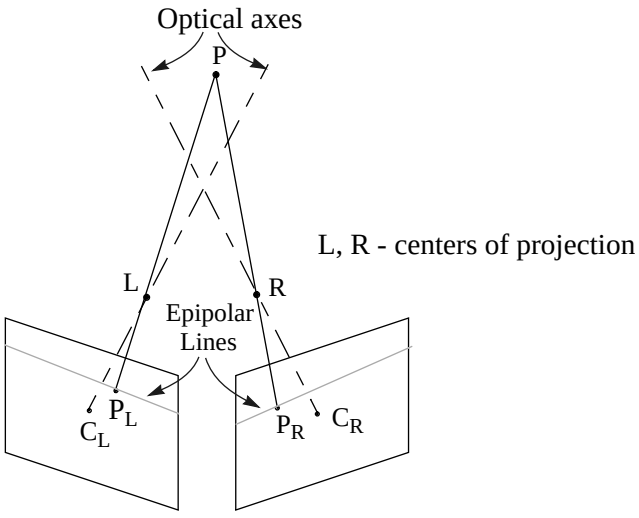


FIG. 2.1: General binocular imaging geometry

(coplanar optical axes)

perspective onto points P_L and P_R on the left and right imaging sensors, respectively, through the left and right imaging elements L and R (pin-hole approximations[1] to the real lenses). The *disparity* of point P, the distance between the *corresponding points* P_L and $P_R$when the two images

are aligned one on top of the other, is inversely proportional to its distance from the centers of projection. The problem of finding all pairs (P_L,P_R), given the left and right view images, is known as the *correspondence* or *disparity estimation* problem [110]. The search for P_L for a given P_R is in general two-dimensional. However, when the optical axes (which are the perpendicular lines to the imaging planes passing through the respective centers of projection) are coplanar, the corresponding points are constrained by the geometry to lie on epipolar lines, defined by the respective intersections of the two image planes with the plane defined by (P, L, and R) [16]. Thus, the search for the corresponding point P_L in the left image, for the point P_R, is restricted to one dimension. In the particular case of the optical axes being parallel (Fig. 2.2), the epipolar lines become the corresponding horizontal scan lines. Hence there is no need to compute the epipolar line in this case. Owing to the presence of *occlusions*, areas that are visible in one view and not in the other, not all image points have a correspondence.

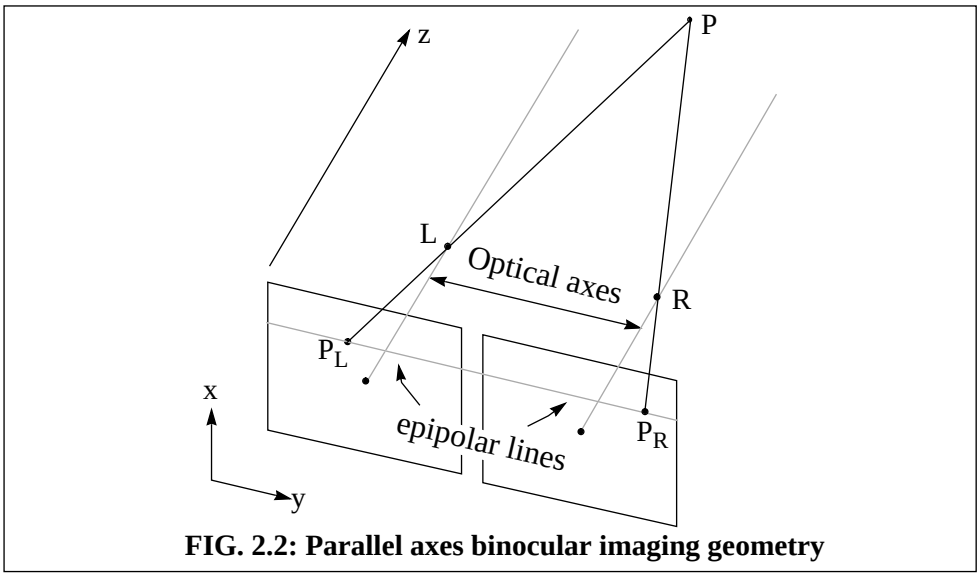


FIG. 2.2: Parallel axes binocular imaging geometry

The proper stereoscopic imaging geometry for stereo viewing is closely related to the stereoscopic display geometry which involves the position of the left and right display screens with respect to the viewer and the angles of view the display screens subtend at the respective eyes. In addition to being a computationally favorable setup, the parallel axes geometry is known to be the right geometry for stereoscopic viewing when the images are displayed on coplanar display screens. This is because the two views do not have any vertical disparity between corresponding points, which the eyes strain to correct for. When the same display screen is used for displaying both views, the geometry places additional restrictions on how to position the image sensor planes relative to the lenses [108].

1. A pin-hole approximation is the model that the imaging element is an infinitesimally small hole and that the image of a point P in the real world, on the image plane, is given by the intersection of the image plane and the line joining P and the pinhole.

2.1.5 Multi-view stereoscopy and intermediate view synthesis

A stereoscopic image pair provides the relative depth information in the actual scene only from one pair of viewpoints. Hence, there is only one correct viewing position. Thus, two views are suited for one viewer and one location only. To allow multiple viewers to see the correct perspective and to provide a single viewer with motion parallax cues during head movement, more than two views are needed. Since it can become prohibitive in terms of acquiring, processing, and transmitting a continuum of views, it would be preferable to acquire only a minimal set of views and to use the knowledge of the relative positions of the acquiring cameras and an estimated disparity map to synthesize views in between two real cameras. Thus, intermediate view synthesis can be considered as a form of compression. However, the quality of the synthesized view depends on the accuracy of the estimated disparity map and the way in which occlusions are handled. But, the disparity estimation can be made more reliable with increasing number of views by using the multiple camera baselines. Typically, a set of cameras on a line, with equal distances between them, is used to acquire the multiple views. These two topics are discussed in detail in Chapter 5.

2.2 DIGITAL VIDEO COMPRESSION

A digital video is typically obtained from an analog video source by digitizing the analog video signal[1]. The luminance (Y) and two chrominance (U and V) components of the analog video signal are first sampled at a sufficiently high rate and then are quantized to a certain precision based on the number of bits allocated for quantization. A single sample (with the three components) is called a picture element, *pixel* or *pel*. Digital video is preferred over analog video because of important reasons such as, (1) further processing on digital video can be carried out at a pixel-by-pixel level; (2) digital transmission can be made more robust than analog transmission; (3) digital copying does not degrade the quality of the copy; and, (4) finer trade-off between perceived quality and transmission bandwidth becomes possible.

2.2.1 Need for compression of digital video

A nominal NTSC [2] video signal has 480 active scan lines per frame and each scan line is typically digitized to 720 pixels. According to the 4:2:2 sampling[2] of the Y, U and V components and a precision of 8 bits-per-component-per-pixel, a digitized NTSC signal would require about 166 Megabits-per-second (Mbps) for the NTSC rate of 30 frames-per-second (fps). This poses a serious problem in terms of transmission and storage. For instance, the original analog NTSC signal that required a bandwidth of 6 MHz now requires 83 MHz, assuming a digital modulation scheme that has a 2 bits/Hz performance. Similarly, about 1 Gigabyte of storage is required for just

1. With CCD imaging technology, it is possible to obtain a discrete readout directly from the CCD sensors, which when quantized would produce the digital video signal.
2. The U and V components are horizontally subsampled by a factor of 2.

50 seconds of raw digital video. Hence, for digital video transmission and storage to become practical, it has to be compressed by at least 1-2 orders of magnitude.

2.2.2 Factors enabling compression

Compression of digital video relies on information theory principles and on psychophysical models of the human visual system (HVS). Natural images typically contain a lot of spatial and temporal similarity. In other words, neighboring pixel values are likely to be highly correlated in space and over time. By decorrelating the images based on a suitable model, it becomes possible to code at a bit-rate close to the actual *entropy*[1] of the source. Such *redundancy removal* leads to *lossless* compression methods. However, lossless compression can typically achieve only very low compression factors[2]. Source coding with respect to a fidelity criterion [84] can be used to increase the compression ratio while restricting the distortions introduced to be below certain permissible limit. Quantization of source symbols with respect to the mean squared error fidelity criterion is a commonly employed source coding method. The properties of the HVS are employed to achieve further compression while providing an almost *perceptually lossless* (albeit numerically lossy) compressed video by removing *perceptual irrelevancies*. A well known irrelevancy reduction example is the subsampling of the chrominance components which takes advantage of the low spatial resolution of the human eye in the color channels. Coding methods that exploit only spatial redundancy are called *intraframe* coding (or simply intracoding) methods and those that exploit temporal (frame-to-frame) redundancy are called *interframe* coding methods.

2.2.3 Waveform-based coding methods

Waveform-based coding methods are based primarily on the statistical properties of the image intensities. These methods do not utilize any a priori or derived information about the physical objects that are present in the scene. These methods are mostly 2D (spatial) and 3D (spatio-temporal) extensions of 1D signal waveform coding methods. Some widely used waveform coding methods are *differential pulse code modulation* (DPCM), *transform coding, sub-band coding, vector quantization* (VQ) and *fractal image coding*.

DPCM based coders employ an M-th order Markov model based causal predictor [1]. Suitable static or adaptive predictive coefficients are derived based on the model; the *residuals* (deviation of the predicted values from the actual values) are coded based on the entropy of the residual symbols[3]. A lossy coding can be carried out by quantizing[4] the residuals[5]. The residuals

---

1. For symbols S_i occurring with probabilities p_i, entropy = $-\sum_i p_i \log_2(p_i)$ bits/symbol.
2. Compression ratio or compression factor is the ratio between the number of bits used to represent an image or sequence before compression and the number of bits needed to represent it after compression.
3. *Entropy coding* is the assignment of code word lengths (in bits) for the symbols depending on their frequency of occurrence, such that the average bits/symbol approaches the entropy of the source.

after prediction typically have a Laplacian probability density function (pdf)[1]. An optimal *scalar quantizer* (SQ) that minimizes the quantization error and the entropy of the quantizer output symbols for such a pdf has been shown to be a uniform quantizer (equal width between the different levels of the quantizer) [1]. Figure 2.3 shows a schematic of a DPCM based lossy coder. Since quantization and coding are done at a pixel-by-pixel level, the average bits per pixel (bpp) cannot be made less than 1. A noncausal predictive coding method is presented in [46].

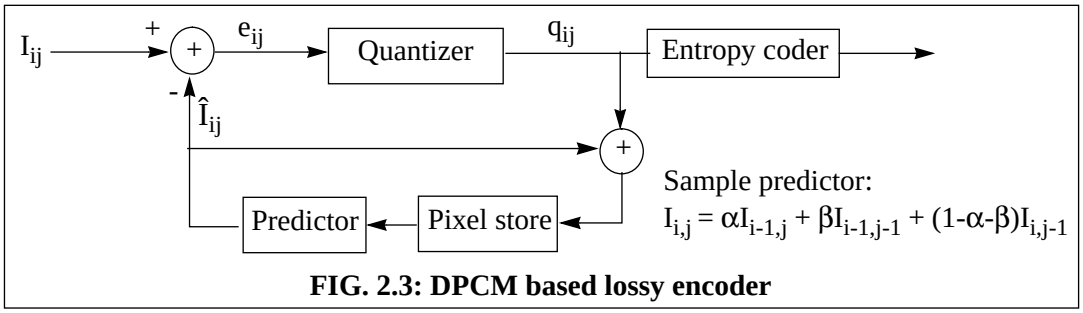


FIG. 2.3: DPCM based lossy encoder

Orthogonal transform based coding methods [8] achieve decorrelation by packing most of the energy[2] present in a signal to be coded into the least possible number of coefficients. Due to the non-stationary nature of images, the transform is usually performed over small rectangular sets of pixels called *blocks*, and the transform basis functions are also usually chosen to be independent of the image data. The *discrete cosine transform* (DCT) has been found to have the most energy packing efficiency for typical natural images [7]. These methods enable irrelevancy reduction better than DPCM methods. For example, the human contrast sensitivity is low at high spatial frequencies[3]. This fact is exploited by quantizing the different frequency coefficients according to the respective contrast sensitivity thresholds (CST) at those frequencies [1]. The nonzero values after quantization are scanned in a zig-zag manner (to increase the number of zero values in a row) and are *runlength encoded*. A typical DCT based encoder is shown in Fig. 2.4. Several computationally efficient implementations of DCT are available [47,111].

Sub-band coding methods [48] are yet another class of waveform coding methods that exploit the non-uniform distribution of energy across the different frequency bands. These methods partition the image into different sub-bands, each of which is coded independently according to

---

4. Quantization is the process wherein the coding rate is traded for distortion.
5. Since DPCM relies on spatial prediction, errors would propagate throughout the entire image if the residuals are not coded.

1. Symbol x occurring with a probability $Ae^{-\lambda|x|}$ (with sum of all symbol probabilities equalling 1).
2. Arising from the concept of area under the power spectrum of a signal, the energy of a video signal can be defined as the sum of the squares of the intensities of pixels over a region of interest, using Parseval's relation [9].
3. Human contrast sensitivity function has a bandpass characteristic. The sensitivity is moderately low at low spatial frequencies and very low at high spatial frequencies.

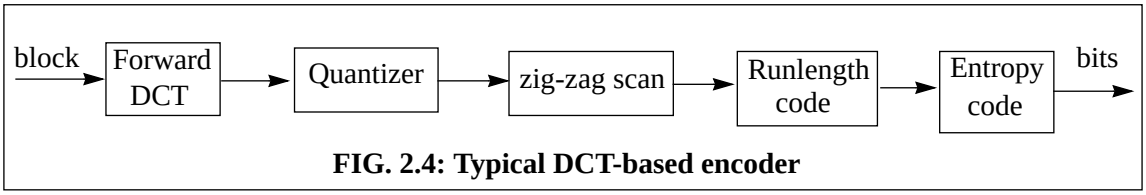


FIG. 2.4: Typical DCT-based encoder

some optimal bit allocation scheme. Since the entire image is filtered and subsampled to obtain the sub-bands, these methods do not suffer from *blocking artifacts* (visible artificial discontinuities across block boundaries) that are common in block-based transform coding methods. Sub-band coding has been shown to be equivalent to coding using an extension of (non-overlapping) block transform called the overlapped or lapped orthogonal transform (LOT) [49]. Further details about sub-band decomposition are presented in Section 2.3.

Vector quantization based coding is an extension of SQ principles to overcome the 1 bpp barrier associated with SQ. Neighboring source pixels are grouped into vectors. Each input vector is coded by finding the closest vector in the minimum *mean squared error* (MSE) sense in a set of vectors called the *vector codebook*; the index of the closest vector in that codebook is coded and transmitted. A vector codebook can be constructed to be optimal for a given image by finding the best set of approximating vectors for the given set of input vectors (called the *training vectors)*. In some cases, vectors in a lattice structure with certain desirable properties are chosen as codevectors, thus obviating the need to transmit the codebook [66, 67, 113]. The most common method for constructing a codebook is the LBG algorithm[1], which iteratively builds the optimal codebook given the number of code vectors [43]. Several computationally efficient VQ methods (such as tree structured VQ) that reduce the search complexity when finding the best approximating code vector and several different variants have been proposed [5]. VQ-based coding can be used for direct image coding [1, 5], residual coding [1, 45], or sub-band coding [48].

Fractal image coding [50] relies on the fact that typical images have self similar structures embedded in them. A set of *domain* blocks with different basic features in them such as simple edges, complex edges and texture are created. In addition, a set of simple geometric transformations (such as reflections about the different axes of the domain blocks and rotations through its center) and massic transformations (linear transformations to account for different contrast and brightness values) are defined. An image to be coded is partitioned into nonoverlapping *range* blocks. Each range block can be further subdivided into up to four sub-blocks, if needed. For each range block, the best matching domain block and the best set of geometric and massic transformation parameters are obtained and coded. In this regard, fractal image coding is similar to vector quantization with a codebook containing all possible combinations of the transformations applied to the domain blocks.

1. The LBG algorithm is the same as the *k-means* clustering method used in pattern recognition [18].

2.2.4 Second generation coding methods

The second generation coding methods are adaptations of waveform coding methods, which however attempt to partition the images into homogeneous regions of arbitrary shapes and sizes depending on some properties, such as texture, color or motion. Hence these methods are also known as *region-based* or *segmentation-based* coding methods. These methods achieve improved coding efficiency over pixel or block based coding techniques by adapting the region sizes according to the local detail present; also these methods improve the perceived quality by reducing artifacts that arise from lumping two non-homogenous areas together (such as blurring of the edge separating the two regions, typical of block based coding). For each segment, the shape, location and the parameters modeling the intensity and color distribution within that segment, need to be coded.

Region growing methods, the early segmentation methods, employ a combination of edge and texture discrimination techniques to obtain homogeneously textured areas [62, 75]. Recently, mathematical morphology has been used to segment images [63, 64]. *Contour coding* is the coding of arbitrary shapes over the discrete grid. *Chain coding* [9], the simplest known way of exactly coding a contour, is typically not bit efficient. The contours can be approximately coded by picking a set of control vertices and by defining a polygon or fitting a spline curve through these vertices. The other option to avoiding arbitrarily shaped regions is commonly known as variable block size (VBS) based segmentation. *Quadtrees* are a well known example of such segmentation [19]. Quadtrees are constructed either in a top-down or bottom-up fashion or as a combination of both. Top-down construction requires recursively *splitting* a block (called a node of the quadtree) into four sub-blocks depending on a splitting criterion. Bottom-up construction requires partitioning the image into small sub-blocks and then recursively *merging* four sub-blocks based on a merging criterion. Split-and-merge techniques construct a top-down quadtree and then merge neighboring subblocks to obtain a collection of subblocks that approximate the underlying arbitrarily shaped region. The tree structure can be coded efficiently with 1 bit per split/merge. However, as the underlying shapes are arbitrary and the blocks are rectangular, the final number of subblocks is typically much higher than with region growing methods. Some extensions to tree-based segmentation that reduce the number of subblocks by allowing diagonal partitions, in addition to horizontal and vertical partitions, are described in [41].

The most commonly used criterion for homogeneity is the intensity variance. The intensity within each segment is usually modeled as a planar (or) quadratic surface, and the parameters of these surfaces are computed by solving the system of equations obtained by applying the model to each pixel in the region [109, 41]. The residuals after fitting the model are coded using conventional methods.

2.2.5 Interframe coding

Image sequences have considerable temporal redundancy as objects in the scene and the camera typically undergo only small displacements between successive frames. Coding methods that exploit this redundancy that exists between temporally adjacent frames are known as interframe coding methods. *Motion compensated prediction (MCP)* is the most widely employed interframe coding method. Even spatio-temporal extensions of transform and sub-band coding methods [2, 61] include a motion compensation stage. In a typical image sequence, motion from frame-to-frame is a composite of the individual object motions and the motion of the camera in the 3-D space, projected on the image plane. The camera motion gives rise to a global motion, while the object motions cause local variations. MCP relies on the fact that, if the local and global motions can be estimated, then a frame (to be coded) can be predicted from a temporally nearby *reference frame*. The error image after prediction, called the *displaced frame difference* (DFD), can be coded using intracoding methods or using segmentation-based coding methods.

Typically the composite local motion is estimated using an approximation to the actual underlying 3D motion model. The region used for motion estimation is typically considered a planar patch that is undergoing motion and a suitable projective transformation is used to model the projection onto the image plane [51, 20]. Translation-only motion parallel to the image plane is the most widely used approximation. This simple model requires only two parameters, namely the horizontal and vertical components of translation. The 2D affine transformation (6 parameter model) is usually a good approximation to the real motion for reasonably distant objects, as it can account for interframe translation, rotation, scaling and shear. The 2D perspective transformation (8 parameter model) is the most appropriate one for modeling the motion of a planar patch under perspective projection; it is thus capable of also accounting for perspective-induced distortions (more noticeable in nearby objects).

Motion estimation (ME) is typically performed for a group of pixels that are likely to have the same motion parameters. ME with a rectangular *block* of pixels and with the translation-only model is commonly known as *block matching* (BM) [52, 53]; it corresponds to finding a block in the reference frame that best matches (in some minimum distortion sense) the block to be predicted. The distortion function is evaluated over a search range centered around the location of zero translation. Though minimum MSE and maximum cross correlation have been used as the criteria for the best match, for computational simplicity the *minimum absolute difference* (MAD) criterion defined below is more widely used.

$$\text{MAD} = Min \sum_i \sum_j \left| I(k, l) - I_{ref}(k + i, l + j) \right| , \quad (i, j) \in S \text{, the search neighborhood.} \qquad \text{(EQ 2.1)}$$

If the distortion function is evaluated at all possible pixel displacements within the search neighborhood, then the search for the best match is called an *exhaustive search*. Since the search

neighborhood can be quite large in real situations, the exhaustive search complexity can be too high to be practical. Several search reduction strategies have been suggested in the literature, based on the assumption that the distortion function is monotonic in the search range. The most notable of these are the *logarithmic search*, *3-step search* and *conjugate direction search* [53]. *Hierarchical block matching* (HBM) is also logarithmically efficient, but it does not make the monotonicity assumption. HBM will be elaborated later in Section 2.3.5. Once the best match in full-pixel displacements is obtained, the estimate can be interpolated to sub-pixel accuracies. The commonly used *bilinear interpolation* [20] uses a linear combination of the four nearest pixels to produce the subpixel value. The horizontal and vertical translatory components together for a block is called a *motion vector.* The motion vectors are usually DPCM encoded to exploit the smoothness of the motion field over the image.

By considering triangular patches and estimating the motion vector at each of the vertices, the six affine model parameters can be obtained. Similarly, the eight parameters of the 2D perspective transform model can be estimated from the motion vectors of the vertices of quadrilateral patches. MCP in these cases proceeds as follows. An image is partitioned into static or adaptive triangular or quadrilateral partitions; the motion vectors of the vertices are estimated using a small neighborhood around each pixel and the affine parameters are computed. The prediction for a patch is obtained by warping the corresponding triangle in the reference frame according to the affine model for that patch. Since motion estimation of vertices may be unreliable, an alternate approach is to iteratively refine the motion model estimates using gradient descent or Gauss-Newton search methods, over the set of pixels within a patch [6].

2.2.6 Model-based coding

These are coding methods that have emerged recently and are a result of the synergy between the three fields namely, image coding, image understanding (scene analysis) and computer graphics (image synthesis). These methods go beyond the 2D intensity information and model the different physical objects in a scene based on their 3D attributes and other available *a priori* information about the scene. Since the images are coded based on their content, these methods are also suited for image indexing and retrieval operations from video databases.

While the conventional coding techniques perform well at high and medium bit-rates, their performance is grossly inadequate at very low bit-rates. This arises partially from the fact that the conventional methods are general-purpose coding methods and are not tuned to take advantage of specific scene types. For instance, in a videoconferencing situation, the camera motion is negligible and the nature of the scene is usually of the ‘head and shoulders’ type. The eye and lip movements are considered most important. However, conventional methods do not take advantage of the nature of the scene and typically allocate bits to all areas of the scene with equal importance. Hence at low bit-rates, the perceived quality is severely degraded. Also, the extent of motion

compensation achieved becomes very important at low bit-rates, as very few bits are available to code undercompensated regions. Hence the conventionally used simple models of motion estimation are to be replaced with more complex models. If the actual objects in the 3D scene and their 3D motion can be modeled, then the sequence can be synthesized from the model parameters by transmitting only the coded object and motion model parameters. This is the motivation behind model-based coding methods. Typically model-based coding methods *track* the objects over time, as opposed to predicting the frame-to-be-coded from a reference frame.

The typical modules of such coding schemes are modeling, image analysis according to the models, model parameter coding, model failure handling, and image synthesis from the models [55, 56, 57]. The analysis stage typically consists of a segmentation[1] stage to obtain the different homogeneous regions in the scene. If the nature of the object is known a priori, as in the videoconferencing situation, suitable 3D surface or volumetric models can be used [60]. Model failure (MF) corresponds to regions that cannot be modeled correctly (such as uncovered background). These regions are usually handled by waveform coding methods [57]. By assuming the objects to be non-rigid and using motion models for flexible objects, the MF regions are considerably reduced. MF areas are also further reduced by allowing geometric distortions (small errors in the size and position of objects) that are more tolerable perceptually than distortion introduced due to inadequate quantization of the MF areas (common at very low bit-rates).

For sequences in which the camera motion is such that it is dominant over object motions and is likely to cover spatially adjacent locations over a long period of time, a new class of methods known as *mosaic-based coding* methods have evolved [82, 115]. These methods register the frames over time using appropriate warping techniques to account for the camera motion and obtain a composite panoramic mosaic image. Thus the temporal redundancies are eliminated. The mosaic is coded using standard intracoding methods. The coded mosaic and registration parameters are sufficient to reconstruct the sequence. Regions with local motion are handled through ‘cut and paste’ operations in [115]. A similar coding scheme that ‘peels’ foreground objects and codes the sequence as a set of layered (depth-wise) constructs is presented in [82].

It should be noted that, because of the a priori knowledge used, a model based coder designed for a particular type of scene is not optimal for coding a different type of scene. Also, the coding efficiency with these coders scales poorly with the number of objects in the scene and as such will continue to be used for less complex scenes only. Some researchers have proposed a switched hybrid coder, that uses model-based coding for coding objects complying to the model and waveform-based coding for coding the model failure regions [59], to encode more complex scenes.

1. The classification into waveform-based, segmentation-based and model-based coding methods is a coarse one and a lot of features are shared.

2.2.7 International video coding standards

Standardization efforts are important for the interoperability of systems built by different product manufacturers; the bit stream compatibility across multi-vendor products also encourages growth and stability in a particular application market. As the demand for digital video in applications such as videoconferencing, digital satellite broadcast, high definition television (HDTV) and internet browsing have increased over the last decade due to increased user acceptance, new standards for image and video coding have emerged. The recommendations by these standards are typically based on the complexity that could be supported by the technology (primarily VLSI fabrication technology) in existence at the time.

In two-way communications, as in videoconferencing, the coder-decoder pair at each end (referred to as a *codec)* has to operate in real-time. The available bandwidth for transmission is also usually limited. The H.261 standard makes recommendations for coding video for such applications [22]. Reduced resolution picture sizes[1] at variable frame rates are adopted to tackle the real-time and bandwidth constraints. The basic unit for processing is a *macroblock* (MB) of size (16 pixels) x(16 pixels)[2]. The chrominance components are subsampled by a factor of 2 in both directions (4:2:0 format). An 8x8 DCT-based intracoding scheme similar to that of Fig.4 and a simple motion compensated interframe coding are employed. The search area for MCP is restricted to ±8 pixels due to real-time constraints. The errors after compensation are DCT coded. Thresholding is used to avoid coding small errors after compensation.

To take advantage of the progress in the video coding field, and the advancements in the VLSI technology, a new videoconferencing standard that is targeted at low bit-rates is currently emerging. This standard-to-be (called H.263 [27]) has several advanced prediction modes. The most significant of these is the overlapped block motion compensation (OBMC) algorithm. The OBMC algorithm permits a different motion vector for each pixel, obtained by interpolating neighboring block motion vectors. This is done to minimize the objectionable blocking artifacts common at low bit-rates.

For more general video coding where there are no real-time encoding constraints, a different standard known widely as the MPEG-1 (Motion Picture Experts Group) standards evolved [24]. This standard focussed on compression for storage media and the target video bit-rate was fixed at 1.2 Mbps (to be within the CD-ROM bit-rate of 1.5 Mbps). Correspondingly a source input format (SIF) of 240 x 352 size frames at 30 fps in 4:2:0 format was chosen. As the encoding need not be done in real-time, the coding scheme is inherently asymmetric in that the encoder is designed to

1. CIF (Common Intermediate format) size of 288 x 352 for bit-rates around 384 Kbps, and QCIF (Quarter CIF) size of 144 x 176 for bit-rates less than 128 Kbps
2. In this thesis, unless mentioned otherwise, the units for size of an image or a block is in *pixels.*

have moderate complexity while the decoder is designed for a low complexity. Some desirable features such as random access to coded frames, editability and the non-real-time encoding, led to a specific frame structure for coding as illustrated in Fig. 2.5.

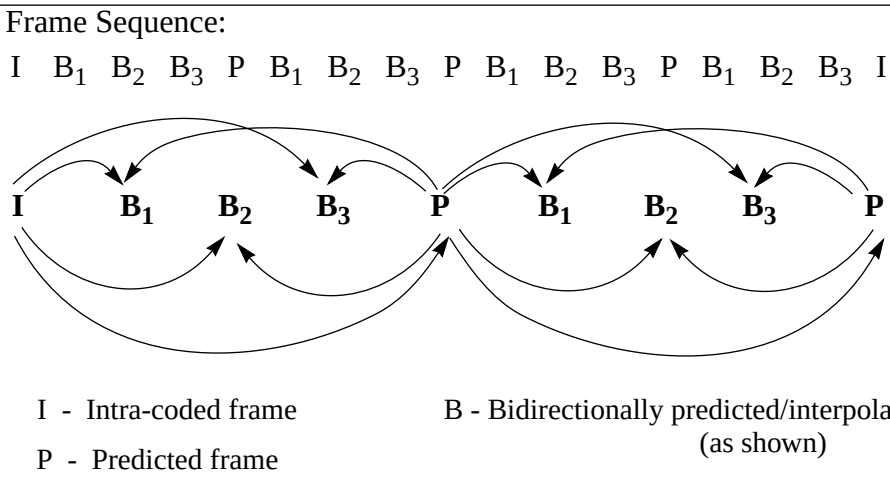


FIG. 2.5: Frame structure in the MPEG standards

Three different types of frames, namely, intracoded (I), predicted (P) and bidirectionally predicted/interpolated (B) frames are used. I-frames are coded using DCT coding of 8x8 blocks; they are periodically placed to enable random access and editability, while preventing accumulation of errors over time. Independent quantization-matrix scale-factors for each macroblock are permitted to facilitate non-uniform bit allocation over a frame. The P-frames are obtained by forward MCP with respect to the previous I or P frame. The B-frames are obtained through forward and backward MCP with respect to the two nearest past and future, I or P frames; the best of the forward match, backward match and an interpolation of the two is chosen. Regions in B-frames that are occluded in one reference frame (due to motion) can be predicted from the other reference frame; also coding errors in B-frames do not propagate. Motion compensation is performed for non-overlapping blocks of fixed size (16x16 or 8x8). A half-pixel accurate motion vector is obtained using bilinear interpolation. The maximum permissible search range is quite high as compared to the real-time codecs. DCT-based coding is used to code the error image after motion compensation.

To handle interlaced video and higher bit-rate video (as in HDTVs) efficiently, a downward compatible standard referred to as MPEG-2 standards evolved [25]. This standard offers different coding modes such as frame-picture-coding and field-picture-coding modes to handle interlaced frames. The motion compensation methods are also suitably modified to have field and frame prediction. Currently, different proposals for a new draft standard called MPEG-4 [28] for very-low bit-rate video coding are being considered. This standard is expected to be based on advanced motion compensation techniques and hybrid model-based coders.

2.2.8 Performance measures

While the performance of a lossless compression method can be evaluated based on the compression ratio alone, the performance of lossy coding methods need to be evaluated in terms of the compression ratio (or coded bit-rate), distortion and perceptual quality. Rate and distortion are objective measures and hence can be computed in a straight-forward fashion. Though compression ratio and coded bit-rate are commonly used to represent the rate, due to subsampling of frames and frame skipping, these measures do not indicate the compression achieved over the actual number of pixels that are encoded. Hence, in this thesis, we will use average bits-per-pixel (bpp) defined below, as the measure for compression efficiency.

$$\text{Average bpp} = \frac{\text{Total number of bits needed to code a stream}}{\text{Total number of pixels in that stream}}$$

The distortion measure that is commonly used is the mean squared error (*MSE*) given by,

$$MSE = \frac{1}{N_x N_y}\left(\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\{I_{actual}(i,j) - I_{coded}(i,j)\}^2\right)$$

Alternatively, the quality is measured using average peak-signal-to-noise-ratio (PSNR) which is related to the mean squared error as follows:

$$\text{Average PSNR (dB)} = 10 log_{10}\frac{255^2}{\left(\frac{1}{N}\sum_{i=1}^{N} MSE_i\right)}$$

where MSE_i is the ith frame's mean squared error and 255 corresponds to the peak intensity level in an 8-bit representation of intensities. Perceptual quality is a subjective measure and requires subjective tests over a large number of viewers. Though subjective tests are difficult to devise and to interpret, they are vital for comparing different coding methods. In addition to these major performance measures, coding methods can also be evaluated in terms of their computational complexity, delay, memory/buffer requirements, spatial/temporal scalability, and resilience to channel errors.

2.3 MULTIRESOLUTION FRAMEWORK FOR VIDEO CODING

A multiresolution (MR) framework is an efficient data structure for image coding that offers several desirable features such as spatial scalability of algorithmic complexity, progressive transmission, and a psychophysical basis for image analysis and representation. The following subsections provide an overview of the MR representation and briefly outline the above features.

2.3.1 Multiresolution decomposition

A multiresolution (MR) decomposition, also known as *pyramid decomposition*, of an image is the decomposition of an image into sub-images at progressively lower spatial resolutions. Such a decomposition facilitates hierarchical refinement of several image analysis methods from a coarse-to-fine spatial resolution level. The decomposition also offers a compact means of coding the image as will be described soon. The coarse-to-fine refinement is computationally efficient and allows spatial scalability. Also, early refinements can be made at a global level, unaffected by local spatial details. Experiments in human visual physiology and psychophysics have shown that the HVS is spatial-frequency selective and that the bandwidth of these spatial filters is likely to be one octave [37]. In other words, the different frequency bands have approximately the same width on a logarithmic scale; this suggests the possibility that the HVS itself employs a multiresolution representation.

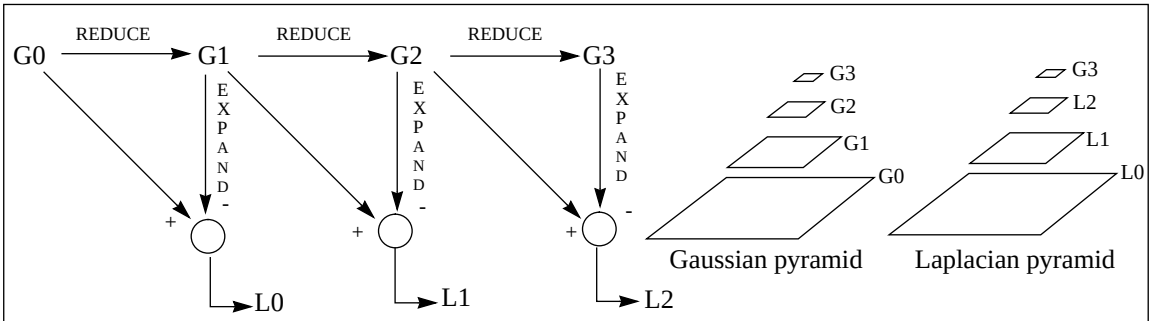


FIG. 2.6: Pyramid decomposition

A decomposition that employs octave-bandwidth filters (followed by subsampling by a factor of 2) to obtain the MR subimages is known as a *dyadic* decomposition. Since a filter with a gaussian shape has compact support in both the spatial and frequency domains, the first proposed MR decomposition [76] used such a filter. However, such a filter does not have unity gain in the entire passband, and hence results in excessive smoothing. The collection of progressively lower resolution sub-images is called a *Gaussian pyramid* and will be used for progressive refinement. By upsampling the image at level-*(l+1)* by a factor of 2 and interpolating using the same low pass filter, a low resolution image with the same spatial extent as the image at level-*l* can be obtained. The difference between these two images that have the same spatial extent, provides the high spatial frequency *details* present at level-*l*. The collection of the detail images at the different resolution levels is called a *Laplacian pyramid*, as the difference of the Gaussian filtered images corresponds to directly applying a Laplacian operator [76]. Figure 2.6 illustrates the construction of the Gaussian and Laplacian pyramids. The coarsest level sub-image of the Laplacian pyramid is the same as the coarsest level sub-image of the Gaussian pyramid. Since the detail images are typically sparse, they can be compressed efficiently. The low pass image contains most of the

energy and can be coded efficiently because of its reduced spatial extent. Thus the Laplacian pyramid constitutes an efficiently codable representation of the original image.

2.3.2 Multirate filter bank theory

While the MR decomposition and the motivation for Gaussian and Laplacian operators arise from vision research, the underlying principles come from multirate filter bank (MRFB) theory in signal processing. MRFB theory presents the framework for the design of suitable filters required in systems that handle different sampling rates. Proper filter design helps achieve important features such as alias cancellation, amplitude and phase distortion reduction, and perfect reconstruction. Thus this theory forms the basis for sub-band decomposition that was discussed in 2.2.3. In sub-band decomposition, an image is decomposed into several nonoverlapping (or minimally overlapping) spatial frequency sub-bands during the analysis stage. Each of these bands can be processed differently. For instance, the human visual system is known to be more sensitive to horizontal and vertical spatial orientations than to other arbitrary orientations. This can be exploited by coarsely quantizing the sub-bands with diagonal orientation. During synthesis, all the processed sub-bands are upsampled and interpolated using the appropriately designed reconstruction filters and added together. For the dyadic decomposition case, the two analysis

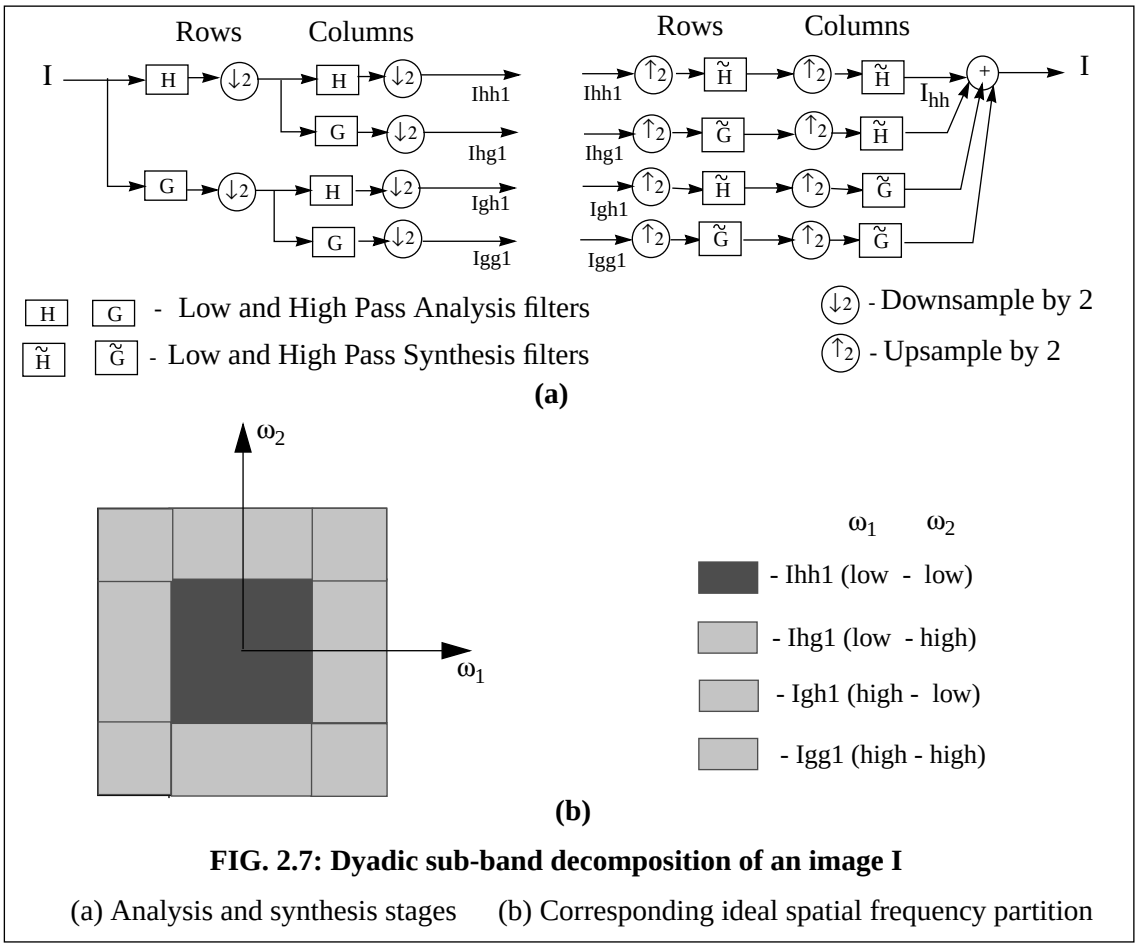


FIG. 2.7: Dyadic sub-band decomposition of an image I

(a) Analysis and synthesis stages (b) Corresponding ideal spatial frequency partition

filters are mirror images of each other w.r.t to the quadrature frequency $2\pi/4$; hence the filters are referred to as quadrature mirror filters (QMF) [11]. Figure 2.7 illustrates the analysis and synthesis steps for a single stage of dyadic decomposition[1] and the resulting ideal frequency partition.The corresponding analysis and synthesis equations are as follows.

$$Ihh1(m,n) = \sum_k h(k) \sum_l h(l) I(2m-k, 2n-l) \quad \text{(EQ 2.2)}$$

$$I_{hh}(m,n) = \sum_k \tilde{h}(2k+i) \sum_l \tilde{h}(2l+j) Ihh1(\left\lfloor \frac{m}{2} \right\rfloor - k, \left\lfloor \frac{n}{2} \right\rfloor - l) \quad \text{(EQ 2.3)}$$

where, *i* and *j* are 0 or 1 depending on whether, *m* and *n* respectively are even or odd. A multiresolution decomposition is achieved by recursively decomposing the low pass subimages only, as shown in Fig. 2.8. The resolution pyramid thus obtained is similar to the gaussian pyramid.

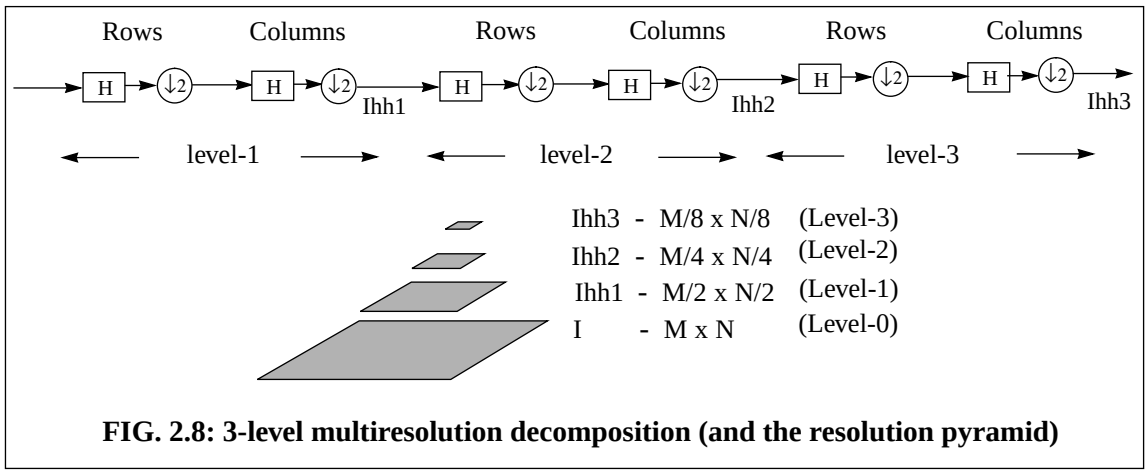


FIG. 2.8: 3-level multiresolution decomposition (and the resolution pyramid)

2.3.3 Wavelet and multiresolution decomposition theory

Wavelet decomposition is a powerful alternative to traditional Fourier analysis techniques for signal analysis. Fourier analysis techniques use basis functions with a fixed spatial (or temporal) support to analyze at all frequencies. Hence good localization in both the spatial and frequency domains is not possible. Wavelet decomposition employs a set of basis functions that are translated and dilated (spatially/temporally scaled) copies of a single function called the *scaling function.* Thus the set of basis functions consists of functions with varying support, hence good localization in both the domains becomes possible. The close relationship between filter bank theory, wavelet analysis and multiresolution decomposition was made popular by the multiresolution

1. A separable 2D filter as shown in the figure yields four sub-bands and is equivalent to the sub-bands obtained after two levels of decomposition using a non-separable 2D filter. However, from a psychophysically based compression point of view, a non-separable filter is considered to be better [].

decomposition theory of Mallat [36]. Filtering equivalences to wavelet decomposition can be found in [36, 72]. The close relationship between the MRFB and wavelet theory provides a rich variety of filter families to choose from, depending on specific requirements. The most commonly used class of wavelet-based filters are the compactly supported orthonormal wavelets of Daubechies [38]. As the name suggests, these filters have a compact support (desirable for computational efficiency) while maintaining a reasonable half-band filter characteristic (needed for minimizing aliasing). The corresponding filter coefficients are derived by applying the "orthonormality under even translations" constraints and regularity constraint (which imposes additional zeros at the sampling frequency to attenuate the high frequency response of the filter). In Section 3.2.1, we list the filter coefficients used in this thesis. The low pass and high pass analysis filters are QMFs and the synthesis filters are just reversed versions of the analysis filters. However, orthogonal filters have an even number of coefficients and are non-symmetric; hence they will have a non-linear phase response. This phase distortion gives rise to varying spatial offsets over the image which may not be acceptable in certain applications that require precise position extraction. A class of symmetric filters with an odd number of coefficients, known as *biorthogonal* filters, have been designed [117] to overcome this drawback. In this case, the low and high pass filters have different lengths.

2.3.4 Laplacian pyramid vs. sub-band decomposition for coding

Though the pyramid and sub-band decompositions are similar in principle, they offer two different representations of the original image. The Laplacian pyramid representation requires four-thirds the number of pixels at the highest resolution level. This increase in the number of pixels is due to the presence of redundancy in the representation. On the other hand, the representation of an image in terms of its sub-bands does not result in any increase in the number of pixels. This is because of downsampling by a factor of 2 in each direction. The aliasing introduced due to non-ideal half-band filters can be cancelled by suitably designing the analysis and synthesis filters. Hence the sub-band decomposition is usually preferred over pyramid decomposition for coding purposes. However, Laplacian pyramid coding has the advantage that quantization errors in the higher levels of the pyramid can be included in the lower level detail images, thus avoiding accumulation of errors [58]. Only the quantization errors while coding the level-0 detail image remain. Such quantization error feedback is not possible in sub-band coding, and the quantization errors can also lead to aliasing during reconstruction. On the other hand, sub-band coding can exploit the orientation sensitivity of the HVS. Methods for independent vector and scalar quantization of the sub-bands are suggested in [68, 69, 70]. The correlation across the sub-bands can be exploited by vector quantizing the vectors formed by the corresponding coefficients in the different sub-bands as in [48]. A zero-tree and successive-approximation based coding of the sub-bands that takes advantage of the predominantly zero regions in the high frequency sub-bands and the correlation across different sub-bands is described in [71]. Both

representations offer progressive transmission capability in which the coarser resolution subimages are transmitted first and the detail images are progressively added. This finds applications in image database browsing as users can download the coarse images first, and, if necessary, can download the detail images later, thus saving considerable bandwidth. Also, in error-prone transmission channels, the coarse subimages that are more critical can be protected with error-correction codes. In this respect, the MR decomposition also allows *prioritization* of information.

2.3.5 Hierarchical block matching on the resolution pyramid

As mentioned in section 2.3.1, the multiresolution pyramid enables hierarchical refinement of motion estimates. HBM was introduced in Section 2.2.5 as a computationally efficient block matching technique. Typically, the most computationally intensive module of a video encoder is the motion estimator. An exhaustive search over a search range of $\pm S$ pixels horizontally and vertically requires $(2S+1)^2$ searches. The complexity of each search is proportional to the number of pixels N used in the MAD computation. Some search reduction strategies which assume a unique minimum within the search area were presented in Section 2.2.5. However, due to noise in featureless areas and possibility of periodic patterns, the MAD function over the search range has multiple minima. Hence these search reduction methods are likely to lead to wrong estimates for the motion vector. On the other hand, in HBM the estimation begins at the coarsest resolution level, where the local details have been averaged and only the coarsest details in the image remain. Hence more global features are matched at the coarse resolution levels, and these reliable estimates are refined according to the finer details at the subsequent levels of resolution.

If n levels of decomposition are employed, the search range at level-n is $\pm S/2^n$ and only $(S/2^{n-1}+1)^2$ searches are required at the coarsest level. Since the number of pixels at level-l is $N/4^l$, the per search complexity is also low. At the subsequent levels, the estimates from the previous resolution level can be refined over a $\pm k$ pixel range centered around the estimate. Thus the overall search complexity for an N-pixel block over a search range of $\pm S$ pixels is given by,

$$\alpha N\left\{\frac{1}{4^n}\cdot\left(\frac{S}{2^{n-1}}+1\right)^2+\frac{4}{3}\cdot(2k+1)^2\right\} \qquad \text{(EQ 2.4)}$$

where α is the proportionality constant associated with per-search complexity and the (4/3)N is the asymptotic pixel sum over the pyramid. The ratio of the search complexities for exhaustive search and HBM can be approximately given by,

$$\frac{1}{2^{-4n}+\frac{4}{3}\left(\frac{2k+1}{2S+1}\right)^2} \qquad \text{(EQ 2.5)}$$

Both terms of the denominator are significantly less than 1 for moderate n, large S and small

k. Thus HBM results in a significant reduction in the computational complexity. For a typical example with S=64, n=3, and k=2, the number of computations is reduced by a factor of 445.

In the refinement described above, the number of candidate pixels used for block matching decreases with the resolution. This can result in unreliable matching at the coarse resolution levels as there are fewer features to match within a block. An alternative is to maintain the block size constant at all resolutions. Thus a block in level-*l* will correspond to four blocks in level-(*l*-*1*). Figure 2.9 illustrates such a HBM. In this case, the number of computations per block is the same as in (2.4). Hierarchical extension to any general motion estimation model is presented in [6].

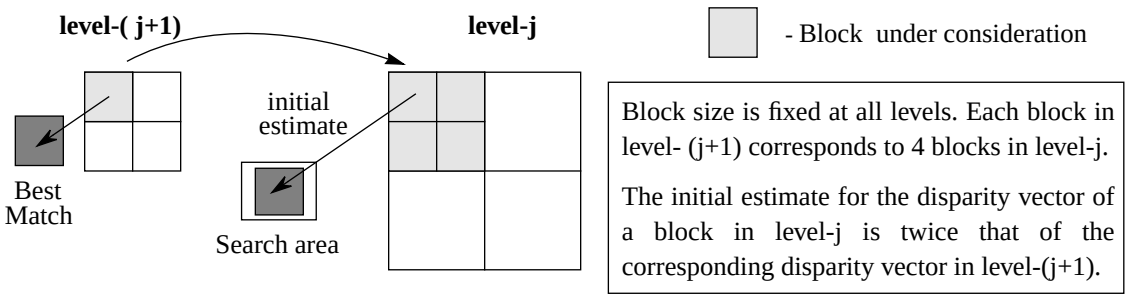


FIG. 2.9: Hierarchical motion or disparity estimation on a dyadic MR pyramid

2.3.6 Other applications of multirate filters in video coding

The interoperability of video encoders and decoders requires handling of a wide variety of display formats. The different television standards such as NTSC, PAL and SECAM that are used in different parts of the world have different display sizes. The proposed HDTV has an aspect ratio of 16:9 and modern movies have an aspect ratio of 3:2, as opposed to the conventional TV aspect ratio of 4:3. Hence, to be able to make maximum use of an available display resolution, an efficient resizing scheme is needed. While the dyadic decomposition provides scaling only by multiples of two, the ratios between these different systems are non-integers. MRFB theory provides an efficient way to handle downsampling and upsampling by different factors. This provides an added incentive to use a multiresolution based approach, so that the same hardware resource can be shared for decoding and display scaling over a variety of display formats. The frame rate difference between different video sources (60 Hz and 50 Hz field repetition rates in TVs and 24 frames per second in movies) can also be handled if the multiresolution concept is extended in the temporal dimension. One such scheme is presented in [58].

Chapter 3

Stereoscopic image compression

In this chapter, we introduce disparity compensated prediction (DCP), which enables one view of a stereo image pair to be predicted if the other view is given. We then briefly outline prior work on this problem, highlight some of the drawbacks with these methods, and articulate the need for a new approach. We then develop a novel disparity-based segmentation (DBS) approach in detail. The computational simplicity of this approach is emphasized through illustrative examples. A model for coding the tree structure and segment disparities is presented; based on these models, a computational scheme to improve the rate-distortion performance of the approach is described. Finally, the performance of the DBS method is compared with that of a fixed-block-size (FBS) based coding method over a test set of stereoscopic image pairs. The segmentation, prediction, and residuals after prediction are shown for a few sample image pairs. The improvement in the disparity map over FBS-based DCP is also illustrated. Multi-view extensions to DBS are deferred until Chapter 5.

3.1 DISPARITY COMPENSATED PREDICTION (DCP)

3.1.1 Introduction

In Section 2.1.4, we introduced the concept of disparity estimation. A stereoscopic image pair is comprised of two views of the same scene from two slightly different perspectives. Hence, barring pixels that are occluded either by scene objects or the frame boundaries, there exists a one-to-one correspondence between the pixels in the two views[1]. This fact can be exploited to predict one view of the image pair given the other view, as shown in Fig. 3.1. However, solving the correspondence or disparity estimation problem is non-trivial. This is because of what is known in the vision literature as the aperture problem. The correspondences can be unreliable if a very small region is considered during the search, whereas including larger areas during search can also lead to erroneous estimates as two objects at different depths are likely to be considered together and a common (erroneous) disparity value is assigned to that region. Thus for different regions of the image, different block sizes that depend on the local disparity detail are needed. Since the local disparity details are not initially available, an iterative estimation of disparity is required. The problem becomes tougher when the correspondences have to be coded. Pixel-wise estimation would require coding the disparity for each pixel. This does not result in good compression.

1. The correspondence is approximate in general, and is exact only in the limiting case of infinitesimally small pixels.

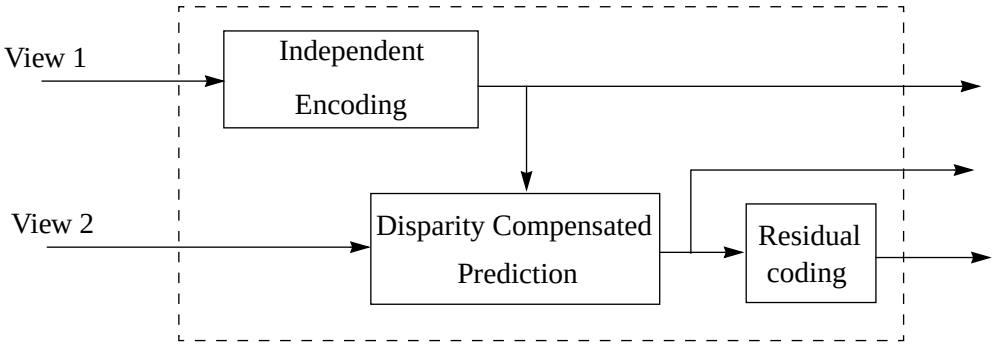


FIG. 3.1: Disparity compensated prediction based coding of a stereoscopic image pair

Hence, disparity estimation methods used for encoding a stereoscopic pair (in contrast to methods used to obtain depth from stereo) typically *assume a constant disparity over a block of pixels*[1]. In this case, the problem is similar to interframe coding methods discussed in Section 2.2.5. However, the major difference in this case is that, due to the epipolar constraints mentioned in Section 2.1.4, the search for the corresponding pixel (or block) is restricted to one-dimension. In contrast, motion estimation requires a 2D search. For the parallel axes stereoscopic imaging geometry, the search for the best match for a block is restricted to be within the corresponding scan lines in the other view. In addition to simplifying the search, this improves the coding of disparities also, as the disparities are scalars in this case.

3.1.2 FBS-based DCP

Several researchers have reported stereoscopic image coding schemes based on DCP. A brief overview of the important works was presented in Section 1.2. Here, we describe some of these methods and point out their shortcomings in light of the objectives set forth in Section 1.3. In [30], a stereoscopic image sequence is modeled as a stationary and ergodic discrete stochastic process that issues two integers from a finite set of integers that represent the set of all possible images (for a given frame size and number of intensity levels). Based on this model, it is shown that the coder structure of Fig. 3.1 provides an optimal coded representation if the images are losslessly encoded. It is also shown that this structure is nearly optimal if the images are coded w.r.t a fidelity criterion. However, the near optimality can be achieved only if the dependence of one view on the other can be fully exploited. The simplistic stoschastic model described above does not provide any such method. From a practical implementation point of view, the paper presents a fixed-block-size based block matching algorithm (FBS-BMA) to estimate disparity. As was discussed in Section 2.2.7, international video coding standards adopt FBS-BMA for motion estimation because of its implementation simplicity. Hence a majority of researchers have applied it for DCP also [31, 87, 89, 91, 97]. However, these methods have certain inherent shortcomings which are

1. Physically, this implies a planar patch that lies parallel to the image sensors at a fixed depth.

presented below.

Typical stereoscopic image pairs have large areas of near-constant binocular disparity. The FBS-based disparity compensation fails to take advantage of such regions, and results in a significantly higher disparity coding overhead than necessary. If the estimated disparity map is smooth, it can be coded efficiently by predictive coding. However, block matching using small featureless areas leads to spurious matches that render the predictive coding of block disparities ineffective. When the fixed size block falls across objects at two different depths, incorrect estimates are produced. Thus the errors after disparity compensation are higher at object boundaries, requiring a higher residual coding overhead. Further, the intermediate views, synthesized based on a disparity map with spurious and incorrect matches, are highly inaccurate.

The number of spurious matches can be reduced by resorting to hierarchical block matching described in Section 2.3.5. HBM for disparity compensation is considered by us in [103]. A very closely related method is described in [99]. These methods improve the smoothness of the disparity map to a certain extent. A genetic algorithm based block matching scheme is described in [91], which specifically tries to achieve a smoother disparity map. However, this method is iterative, and no compression results are presented.

3.1.3 Second generation and model-based disparity estimation methods

Several edge-based methods for solving the correspondence problem have been proposed in the machine vision literature [see 118] and some of these methods have been extended for use in coding applications [33]. These methods typically detect intensity edges by convolving the image with a Laplacian-of-Gaussian operator and extracting the zero crossings. The extracted edges are approximated by straight line segments and labelled. Correspondence is established for an edge in one view by searching for an edge with similar orientation and length in the other view using a suitable optimization method. Dynamic programming methods have been proposed to establish such correspondences [33, 83]. The correspondences at the edges need to be propagated to other pixels. In general, the contour or edge-based disparity estimation schemes are computationally intensive, and are not efficient from the coding point of view.

Recently, model-based image coding methods [100, 99] have been applied to make the disparity compensation adaptive to the actual objects present in the scene. As was explained in Section 2.2.6, these model-based coding methods are well suited only for restricted applications. In general the performance of these methods do not scale well with the number of objects in the scene and with complex camera and object motions. Also the analysis stage in these methods are computationally intensive. Hence these methods cannot be applied for arbitrary scenes. Improvement in coding performance over conventional methods for general imagery has not yet been established.

3.1.4 Motivation for a new approach:

Thus, the computationally simple FBS-based DCP methods do not provide an optimal coding representation. The advanced methods address this problem, but do not perform well for arbitrary imagery. Hence we need a new approach that adapts disparity coding bits to the local disparity detail present in the image, while maintaining a low overhead for coding these adaptive segments and a moderate computational complexity. We maintain that a representation optimal for disparity coding can be obtained by segmenting the stereoscopic image pair based on the disparity. Some earlier attempts at segmenting an image into foreground and background areas for videophone applications using a stereo camera setup are described in [39, 74]. In this subsection, we formalize the need for a such an approach.

Let us assume that a suitable model can be formulated to map a set of pixels in one view of the stereo pair to a corresponding set of pixels in the other view. Let there be N arbitrarily shaped regions such that the correspondence for pixels within each region is specified by the model parameters for that region. Let R_k^{model} be the number of bits needed to code the model parameters for the *k*th region[1]. Let R_k^{shape} be the number of bits needed to code the shape of the *k*th region, either in a lossy or lossless fashion. Since there will be approximations made in the model and in the shapes, there will be errors after prediction using the model. Let R_k^{error} be the number of bits needed to code these errors subject to a fidelity criterion. In addition, there will be regions which, due to occlusion, do not have a corresponding region in the other view. Let R^{occ} be the number of bits needed to code these regions, either by intracoding or by finding a similar region in the other view and coding the residuals. Hence the total number of bits needed to code one view given the other, subject to a fidelity criterion, is:

$$R^{total} = \sum_{k=1}^{N} (R_k^{model} + R_k^{shape} + R_k^{error}) + R^{occ} \qquad \text{(EQ 3.1)}$$

This expression clearly shows the different dimensions that affect the coding performance. It also shows that the coding problem can be optimized by independently optimizing the coding of the unoccluded regions (first term) and the occluded regions. However, precise detection of occluded regions is non-trivial. Hence the occlusions are typically handled in the same way as unoccluded regions, with large errors due to model failure being taken care of at the error coding step.

For the FBS-based methods, $R_k^{shape} = 0$ as the regions are chosen independent of the images. The number of blocks N_{fbs} is typically much larger than N. In addition to the increase in bits due to

1. A different R_k^{model} for the different regions is assumed to loosely account for the fact that the bits for coding the model parameters can be reduced by predictive coding and entropy coding.

the larger N_{fbs}, R_k^{error} also increases for blocks that contain objects at different depths. The previously proposed contour and model based coding methods do not segment based on disparity, and hence typically have a larger number of regions than N. Also, R_k^{shape} is typically very high; hence these methods do not scale with N. In the next section, we propose a new approach that segments based on disparity and minimizes R^{shape} using a new multiresolution based quadtree decomposition method.

3.2 DISPARITY-BASED SEGMENTATION APPROACH

In this section, we present a new approach for efficient disparity compensated coding of stereoscopic image pairs. This approach, which we refer to as disparity-based segmentation, combines intensity and disparity information to segment one view of a stereoscopic image pair given the other, and achieves a coding representation that is commensurate with the local disparity detail. A quadtree decomposition is employed as opposed to contour-based segmentation because the segmentation structure coding overhead scales well with the complexity of the scene. A computationally efficient, non-iterative solution, that reduces the segmentation overhead as well, is obtained by using a multiresolution framework.

The details of the MR framework are outlined in Section 3.2.1. We generalize quadtree decomposition (QTD) from the coding point of view in Section 3.2.2, and explain how the quadtree decomposition is performed within the MR framework. The partitioning locations for the generalized QTD are computed using a simple edge detection scheme described in Section 3.2.3. The manner in which this new approach reduces the overhead needed to code the segmentation structure is explained in Section 3.2.4. The multiresolution based DBS algorithm is presented step-by-step in Section 3.2.5. Compression results are presented in Section 3.2.6. A scheme to optimize the rate-distortion (R-D) performance of DBS by augmenting the splitting criterion with a R-D threshold is outlined in Section 3.2.7. The computational complexity of the DBS scheme and the degree of parallelism offered by it are discussed in Section 3.2.8.

3.2.1 Multiresolution framework for DBS

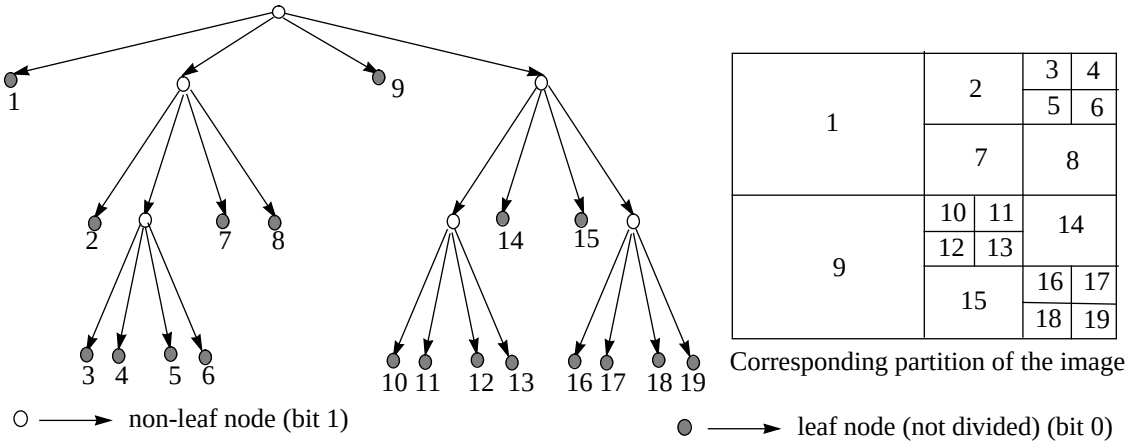
To segment based on the binocular disparity, we need an accurate disparity map. But an accurate, disparity-discontinuity-preserving disparity map can be estimated only from a good segmentation of the image. This implies an iterative solution, which can be computationally prohibitive. However, a MR framework enables us to progressively refine both the partitions and their disparities from a coarse-to-fine resolution, thus significantly reducing the associated computational burden. The framework also provides several additional desirable features: (1) A mixed resolution stereoscopic image coding scheme can be realized with ease within the framework; (2) As we will see in Section 3.2.2, the multiresolution estimation (MRE) enables us

to apply different splitting strategies at different resolutions to reduce the segment information coding overhead; (3) The MRE also reduces spurious matches by avoiding local minima during block matching; (4) In addition, the entire coding scheme becomes scalable with resolution. The estimation accuracy does not depend much on the choice of the analysis filters. We have chosen Daubechies' compactly supported 6-tap filter as it offers a reasonable half band frequency response with a small number of coefficients [38]. The filter coefficients are given below.

h(1) = 0.23523360389202 h(2) = 0.57055845791566 h(3) = 0.32518250026277
h(4) = -0.09546720778398 h(5) = -0.06041610415518 h(6) = 0.02490874986582

3.2.2 Generalized quadtree decomposition

Quadtree decomposition of an image is a structured recursive partitioning of an image into rectangular blocks based on a splitting criterion. Figure 3.2 illustrates a typical quadtree. At each level of the tree, the blocks are referred to as *nodes* of the tree and the nodes that are not partitioned further are referred to as the *leaf nodes*. Typically, a block is divided only at the midpoints of its sides. In such a *regular decomposition*, the tree structure and thus the size and location of each node can be represented using only *1 bit/node*. Hence the overhead needed to represent the tree structure, referred to as the segmentation overhead, is very minimal. However, as the partitioning location is obtained independent of the features within the image, the regular decomposition typically results in a larger number of blocks.



(Leaf nodes are numbered sequentially in the depth-first traversal order)

The order in which the four subblocks within a block are processed is:
top-left, top-right, bottom-left and bottom-right

10101000000011000000010000 - Encoded tree structure (depth-first tree traversal)

FIG. 3.2: Illustration of regular quadtree decomposition

Spatial homogeneity of a block, e.g. [41, 109], and block motion, e.g. [42], have been used as

splitting criteria in the literature. In this thesis, we propose a novel partitioning criterion. Since typical scenes have large regions that are nearly at a constant distance from the camera, an object oriented segmentation of the scene can be obtained by using the disparity or depth of a block as the splitting criterion. Hence the number of block disparities to be coded after DCP is considerably reduced for typical scenes. Use of regular decomposition would decrease the segmentation overhead, but would increase the final number of blocks after decomposition and hence would increase the number of bits needed to code their disparities. The objective, then, is to minimize the *total* number of bits required to code the quadtree structure and the block disparities. The number of leaf nodes can be minimized by aligning the partitioning location with disparity discontinuities. However, coding the locations of the arbitrary horizontal and vertical partitions within a block requires log_2*(size of the block) bits/node*. Instead of creating four subblocks always, the number of leaf nodes can be reduced by considering horizontal (H) and vertical partitions (V) independently. This would, however, require *2 bits/node* to code the four cases, namely, H only, V only, H and V, and neither H nor V.

We consider a *generalized* quadtree decomposition to address the trade-off between the number of leaf nodes and the segmentation overhead. A block can be divided horizontally and vertically at 2^K-1 locations that are evenly separated. K is the number of bits allocated per direction per node to specify the partitioning location. The division takes place at the permitted location that lies closest to a sharp disparity discontinuity. Since disparity discontinuities are not available before the segmentation, intensity edges which usually constitute a super-set of disparity discontinuities are used. Figure 3.3 illustrates the generalized quad-tree decomposition procedure.

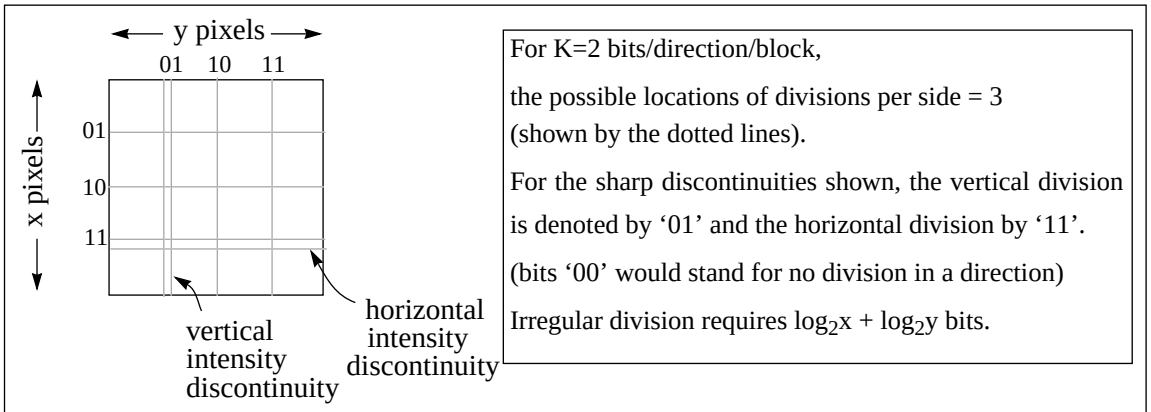


FIG. 3.3: Generalized quadtree decomposition - partitioning locations for K=2

The regular decomposition corresponds to (K=0) and irregular decomposition corresponds to (K = length/width of the block[1]).

1. Throughout this thesis, unless indicated otherwise, the units are in *pixels* at the respective resolution levels.

A multiresolution based QTD proceeds from the coarsest resolution to the finer resolution levels. The leaf nodes at one resolution become the root nodes at the next resolution level. This unique MR framework for QTD greatly simplifies the complexity of the decomposition and also helps in minimizing the coding overhead. For instance, at the top of the tree, if the disparity compensation is performed at the original resolution, the search has to be conducted for block sizes that are close to the size of the image itself, whereas, with the MR framework, the estimation is performed at a coarser resolution level. By employing different values of K at the different resolution levels, the segmentation overhead and the total disparity coding bits needed can be jointly minimized. For instance, larger K values can be chosen at the coarser resolutions as there will be fewer blocks initially. Regular partition can be used at finer resolutions, as the number of blocks is higher and the error due to fixed partition is smaller at these resolutions due to the smaller block sizes. Since the disparities can be differentially encoded on the tree, the required number of disparity coding bits are also reduced (see Section 3.2.7).

3.2.3 Partitioning location calculations

The primary objective of an irregular decomposition is to align the block boundary with the boundary of the attribute that is used as the splitting criterion. In our case, this boundary is the disparity discontinuity. Disparity discontinuity which arises from an object boundary typically lies at an image intensity discontinuity (edge). In the absence of a disparity map (which is what we are trying to estimate), the edges in an image provide the most practical candidate locations for partitioning. Conventional edge detection requires convolution of the block with two gradient operators (such as a Sobel operator) in orthogonal directions. The intensity gradient at each pixel is then thresholded to obtain an edge map. The 2D convolution with the operators is computationally expensive. Also, we need only the dominant horizontal and vertical edges within a block. Hence we use a computationally simple dominant vertical and horizontal edge locating algorithm. For a block of size w x h starting at location (x,y) in image I, the row and column means are computed as,

$$m_{row}(i) = \sum_{j=x}^{x+w-1} I(i, j) \quad \text{and} \qquad \text{(EQ 3.2)}$$

$$m_{col}(j) = \sum_{i=y}^{y+h-1} I(i, j) \; . \qquad \text{(EQ 3.3)}$$

These averages provide us with two 1-D signals. The effect of local details and noise are averaged out and the dominant edges along the horizontal and vertical directions become emphasized in the row and column averages. A symmetric difference high pass filter is applied to the row and column averages. By finding the peak over the absolute values of the filter outputs, the

horizontal and vertical partitioning locations are computed as follows:

$$\text{Horizontal: } l_h = \underset{i}{Max}\,(g_h(i) = \left|(m_{row} \otimes f)\right|) - n; (0 \le i < h + 2n) \quad \text{from } x \qquad \text{(EQ 3.4)}$$

$$\text{Vertical: } l_v = \underset{j}{Max}\,(g_v(j) = \left|(m_{col} \otimes f)\right|) - n; (0 \le j < w + 2n) \quad \text{from } y \qquad \text{(EQ 3.5)}$$

where operator $\otimes$ denotes discrete convolution and *f* is a symmetric difference filter of length *(2n+1)*. The typical filters we use are of orders *n*=1 and *n*=2 (specifically, [-1, 0, 1] and [-1, -2, 0, 2, 1]). A larger *n* provides a more reliable edge location by smoothing out local variations, but reduces the number of candidate partitioning locations due to edge effects.

This procedure is illustrated for a test image in Fig. 3.4. It can be seen that good candidate locations that are well aligned with intensity discontinuities are obtained using this computationally simple procedure.

3.2.4 Segmentation overhead coding

While at first glance it appears that the segmentation overhead is considerable for irregular decomposition, by imposing limits on the maximum and minimum allowable block dimensions, and by operating within the MR framework, the overhead can be reduced considerably.

If the width and height of a block are *w* and *h* respectively, then coding the locations of an arbitrary vertical and horizontal partition requires $log_2(w.h)$ bits. Since the dimensions of the blocks progressively get smaller as the quadtree decomposition proceeds, the number of bits required to code the locations decreases logarithmically.

If *n*-levels of dyadic MR decomposition are employed, then the size of the sub-image at the coarsest resolution is 4^{-n}th of the actual image size. Thus the partitioning location coding bits for a block that has w x h dimension at the full resolution, requires only $[log_2(w.h) - 2n]$ bits at the *n*th resolution level. By using large K values at the coarser resolutions (where the above fact can be exploited) and smaller K values at the finer resolutions as described in Section 3.2.2, the coding overhead can be considerably reduced.

Further, if the maximum and minimum allowable block dimensions are set to be S^n_{max} and S^n_{min}, respectively, at the *n*th resolution level, then from the *(n-1)*th level onwards the upper bound for coding overhead per partition will be $log_2(2S^{k+1}_{max} - 2S^k_{min})$ at the *k*th resolution level. The S_{max} values aid in upper bounding the complexity that needs to be handled by a processing element in a parallel processing implementation (see discussion in Section 3.2.8). The S_{min} values prevent the formation of extremely small blocks and also account for the inaccuracy at the edges of

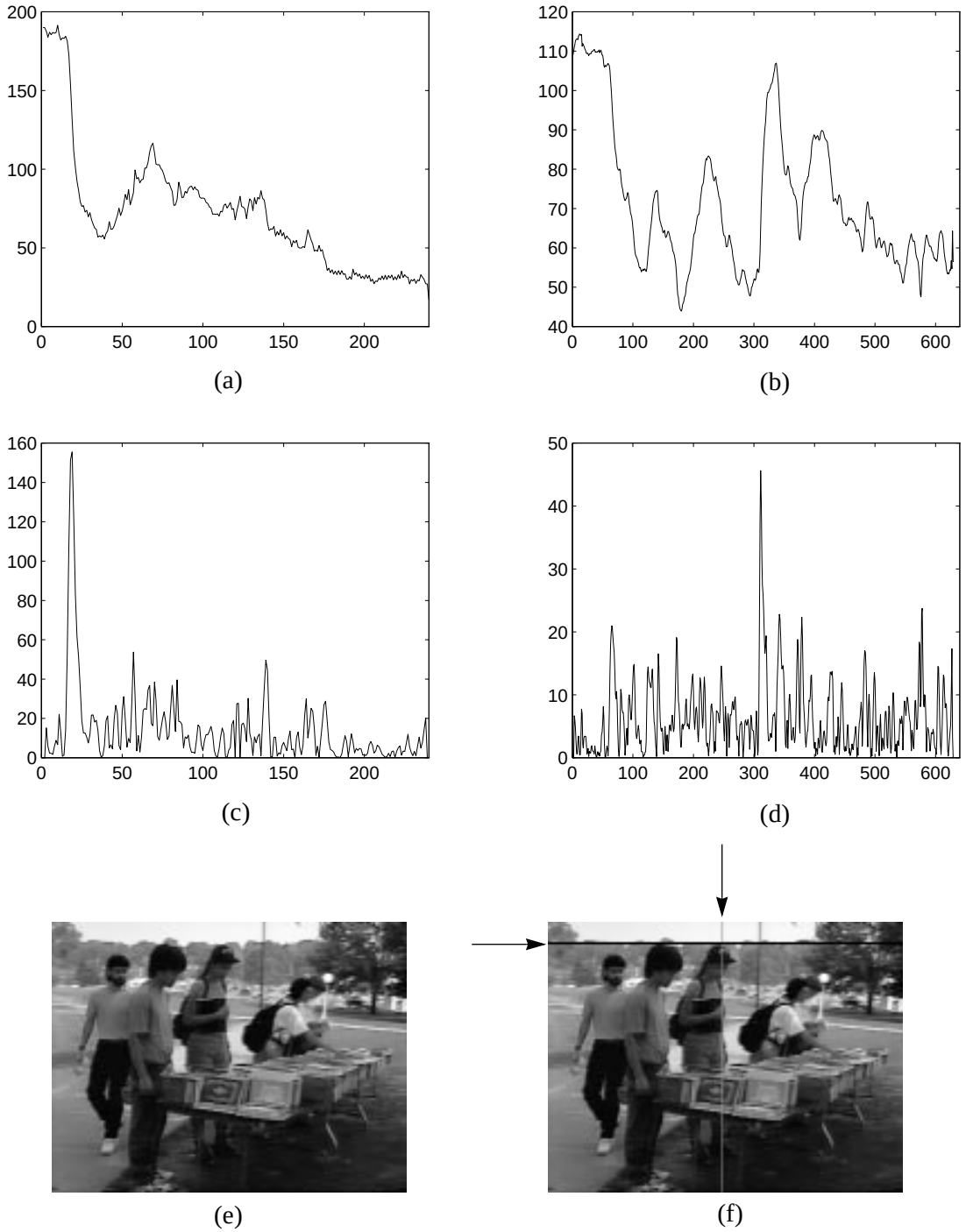


FIG. 3.4: Illustration of partitioning location calculation

(a) Row averages (b) Column averages (c) Absolute values of the high pass filtered row averages (d) Absolute values of the high pass filtered column averages (e) Image used (f) Image with the horizontal and vertical partitions which correspond to the maxima in (c) and (d) (the strongest horizontal and vertical intensity discontinuities over the block).

The symmetric difference high pass filter used is [-1 -2 0 2 1].

a block while using the symmetric difference filter in the last section.

Since the same number of bits would be needed to code the arbitrary partitioning location irrespective of whether a block is split or not, the tree structure is coded in two separate levels. At the first level, a 2-bit overhead per node is used to specify whether a node was not split, or was split, horizontally, vertically, or both horizontally and vertically. As the block sizes can be computed at the decoder, this overhead can be made to range from 0-2 bits depending on whether the height or width of a block is less than or greater than S_{min}. The partitioning locations are coded at the second level, only in the direction in which the split occurs.

Advantages of the irregular decomposition over regular decomposition is illustrated in Fig. 3.5, for a synthetic test image. The segmentation overhead is almost the same for both decompositions. However, if the disparity of each leaf node has to be coded, then the irregular partition would clearly outperform the regular decomposition. The general equations describing the model for coding the segmentation overhead are developed in Section 3.2.7.

3.2.5 Disparity-based segmentation algorithm

Within the MR framework, different splitting criteria can be used at the different resolution levels. To obtain a reasonable initial segmentation, and to avoid performing block matching with the large blocks at the start, a spatial-homogeneity based decomposition is employed at the coarsest resolution level. The spatial-homogeneity of a block is measured in terms of the intensity variance within the block. At the subsequent resolution levels, the thresholded disparity difference between sub-blocks is made the splitting criterion. The steps of the algorithm are described below.

1. Construct the left and right multiresolution pyramids by recursively low pass filtering and then subsampling using the method shown in Fig. 2.8.

2. Start at the coarsest resolution level with the entire subimage as a block. Set a threshold on the maximum variance (T_{max}) allowed within a block. Set the maximum and minimum permissible block dimensions (S_{max} and S_{min}) at the current resolution.

3. Recursively, for each block of height h and width w:

   If (($h < S_{min}$) and ($w < S_{min}$)), then declare the block as a leaf node.

   Else,

   a. Compute the variance *var* of the block.

   b. If the ($var < T_{max}$) and $(h < S_{max})$ and $(w < S_{max})$, declare the block as a leaf node.

      Else, compute the dominant horizontal and vertical edge locations (l_h and l_v pixels respectively from the top left corner of the block) as discussed in Section 3.2.3.

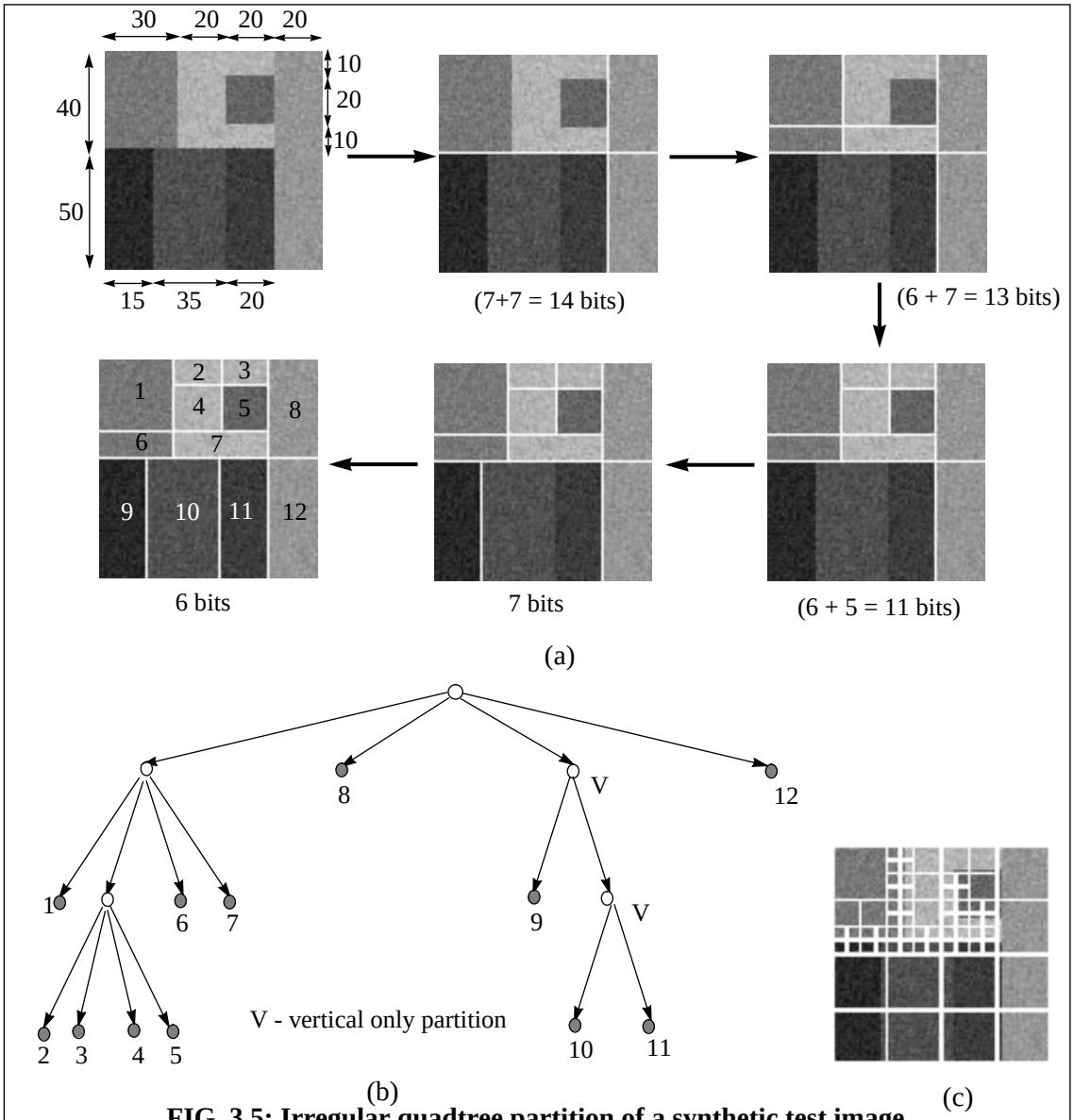


FIG. 3.5: Irregular quadtree partition of a synthetic test image

(a) Steps of the decomposition - no. of bits needed to code the partitioning locations is shown for each step (b) Corresponding quadtree structure (c) Regular quadtree decomposition for comparison.

The number of steps in (a) corresponds to the number of non-leaf nodes in (b).

Leaf nodes are numbered in depth-first order in (b) - corresponding blocks are shown in (a).

As all the edges are either horizontal or vertical in the synthetic image, the partitioning locations computed using the scheme in Section 3.2.3 are aligned exactly with the intensity edges.

	Regular decomposition	Irregular decomposition
Number of leaf nodes	67	12
Partition location coding bits	x	51
Tree structure coding bits	86	34

If motion or disparity for each leaf node needs to be coded, the balance tilts clearly in favor of irregular decomposition.

If (($h - l_h > S_{min}$) and ($l_h > S_{min}$)), divide the block horizontally.

If (($w - l_v > S_{min}$) and ($l_v > S_{min}$)), divide the block vertically.

4. For the leaf nodes at that resolution, compute the block disparity by block matching with the corresponding sub-image at that resolution in the other view. If n-levels of dyadic decomposition are employed, then the search range at the coarsest resolution will be 2^{-n}th of the search range desired at the highest resolution, in the horizontal and vertical directions.

5. Proceed to the next higher resolution level. Double each of the leaf node block dimensions and the corresponding block disparities. Set a threshold for the maximum allowable absolute difference in the block disparities (D_{max} - typically small) between sub-blocks. Set the maximum and minimum permissible block dimensions (S_{max} and S_{min}) at the current resolution.

6. Recursively, for each block of height h and width w:

   a. If ($h > S_{min}$), compute the dominant horizontal edge location l_h.

      If (($h - l_h > S_{min}$) and ($l_h > S_{min}$)), permit horizontal division.

      If ($v > S_{min}$), compute the dominant horizontal edge location l_v.

      If (($w - l_v > S_{min}$) and ($l_v > S_{min}$)), permit vertical division.

   b. For each of the possible sub-blocks in step (a), compute the block disparities[1]. The search range is independent of the resolution level (say, $\pm$ 2 pixels around the current estimate). If the mean absolute error (*MAE*) after compensation is above a preset threshold, the current estimate is ignored and block matching is performed again with the search range at level-l set to 2^{-l}th of the search range at level-0. This is done to prevent the propagation of wrong estimates down the pyramid.

   c. If (the difference between the sub-block disparities > D_{max}) or ($h > S_{max}$) or ($w > S_{max}$), divide the block at the locations determined in step (a).

      Else, declare the block as a leaf node.

7. If the current resolution level is the highest resolution level, then compute half-pixel accurate disparities for the leaf nodes.

   Else, go to step 5.

---

1. The dominant edge is ignored during disparity estimation and is assigned to the sub-block with a larger disparity. Since an edge at an object boundary belongs to a foreground object, the above step prevents this edge from being erroneously assigned to a background object and improves estimation accuracy.

3.2.6 Results

The DBS algorithm was tested on several stereoscopic image pairs selected from a wide variety of sources. Appendix-A presents the details about how these images were acquired. In Table 3.1, the total bits-per-pixel (bpp) including the segmentation overhead (R_{seg}) and the disparity coding bits (R_{disp}) for the DBS algorithm (R_{dbs}) is compared against the disparity coding bits for a FBS-based algorithm (R_{fbs}) that uses 8x8 blocks, at nearly equal PSNRs. The disparity

TABLE 3.1: Compression ratio comparison between fixed-block-size-based DCP and disparity-based segmentation at similar PSNRs

Stereoscopic image pairs	FBS (8x8)			DBS					$\frac{R_{fbs}}{R_{dbs}}$ x100
	No. of blocks	PSNR (dB)	R_{fbs} bpp	No. of blocks	PSNR (dB)	R_{seg} + R_{disp} = R_{dbs} bpp			
Booksale (frame 68)	2400	28.02	0.112	775	27.49	0.023	0.044	0.067	59.8
Crowd (frame 131)	2400	27.29	0.117	766	26.67	0.022	0.049	0.071	60.7
Aqua (field 40)	3240	24.79	0.120	1147	24.30	0.028	0.051	0.079	65.8
Train (field 60)	3240	27.44	0.116	869	27.46	0.022	0.038	0.060	51.7
Tunnel (field 120)	3240	26.11	0.119	1271	25.93	0.030	0.055	0.085	71.4
Piano (field 20)	3240	26.08	0.110	703	25.68	0.019	0.031	0.050	45.4
F. Garden (frames 1&4)	1320	24.63	0.088	437	24.47	0.0201	0.0348	0.055	62.5
Lake	3364	27.79	0.12	690	27.64	0.0203	0.033	0.053	44.1
Group-photo	3564	28.28	0.112	812	27.60	0.0201	0.0302	0.050	44.6

vectors are coded by differential prediction followed by entropy coding (similar to motion vector encoding in MPEG-2 Test Model [26]). It can be seen that the DBS algorithm results in a bpp which is 45-75% of the bpp required for a FBS algorithm. The PSNRs for the DBS case are slightly less than the PSNRs for the FBS case. This is because large blocks tend to have small uncompensated areas that are too small to affect the disparity estimation (or) are allowed as a result of imposing an S_{min}. Also, the constant disparity assumption is less valid as the block size increases. However, these errors are typically concentrated at object edges and thus do not produce

any visually displeasing artifacts.

Figure 3.8 shows the trade-off between PSNR and R_{dbs} that can be achieved for (arbitrarily

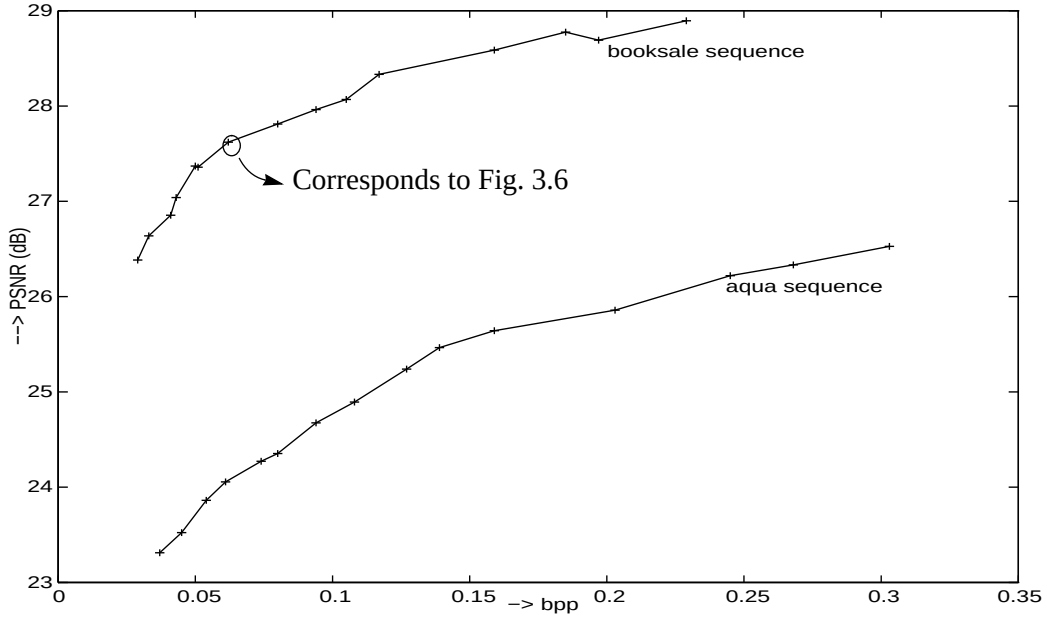


FIG. 3.8: Rate distortion curves obtained by varying the DBS parameters

chosen) stereoscopic image pairs from the *booksale* and *aqua* sequences (see Table 3.1 for the specific frame or field numbers) by varying the S_{min} and D_{max} parameters at the different resolution levels. As the S_{min} and D_{max} parameters are made smaller, R_{dbs} increases due to the increase in the number of blocks. Due to better disparity compensation, the PSNR also increases. However, as S_{min} and D_{max} are reduced to very small values, smaller blocks are created at the lower resolutions and the reliability of block matching decreases. Also, occluded regions tend to get divided into several tiny blocks resulting in an unjustified increase in R_{dbs} with no appreciable improvement in PSNR, as shown by the levelling-off of the curves in Fig. 3.8.

The multiresolution based update of the segmentation and disparity maps for stereoscopic image pairs from the *booksale* and *tunnel* sequences (see Table 3.1 for the specific frame or field numbers) are shown in Figs. 3.6 and 3.7 to illustrate the DBS algorithm. It can be seen that the DBS algorithm results in a smoother and more accurate disparity map compared to FBS based DCP (with 8x8 blocks). The views predicted using DBS and the prediction residuals are also shown. It can be seen that most of the significant residuals are present in the regions that are occluded in the reference frame.

3.2.7 Optimizing the rate-distortion performance

In Fig. 3.8, we showed the R-D performance achieved by varying the multi-level segmentation parameters. However, varying the segmentation parameters alone will not result in

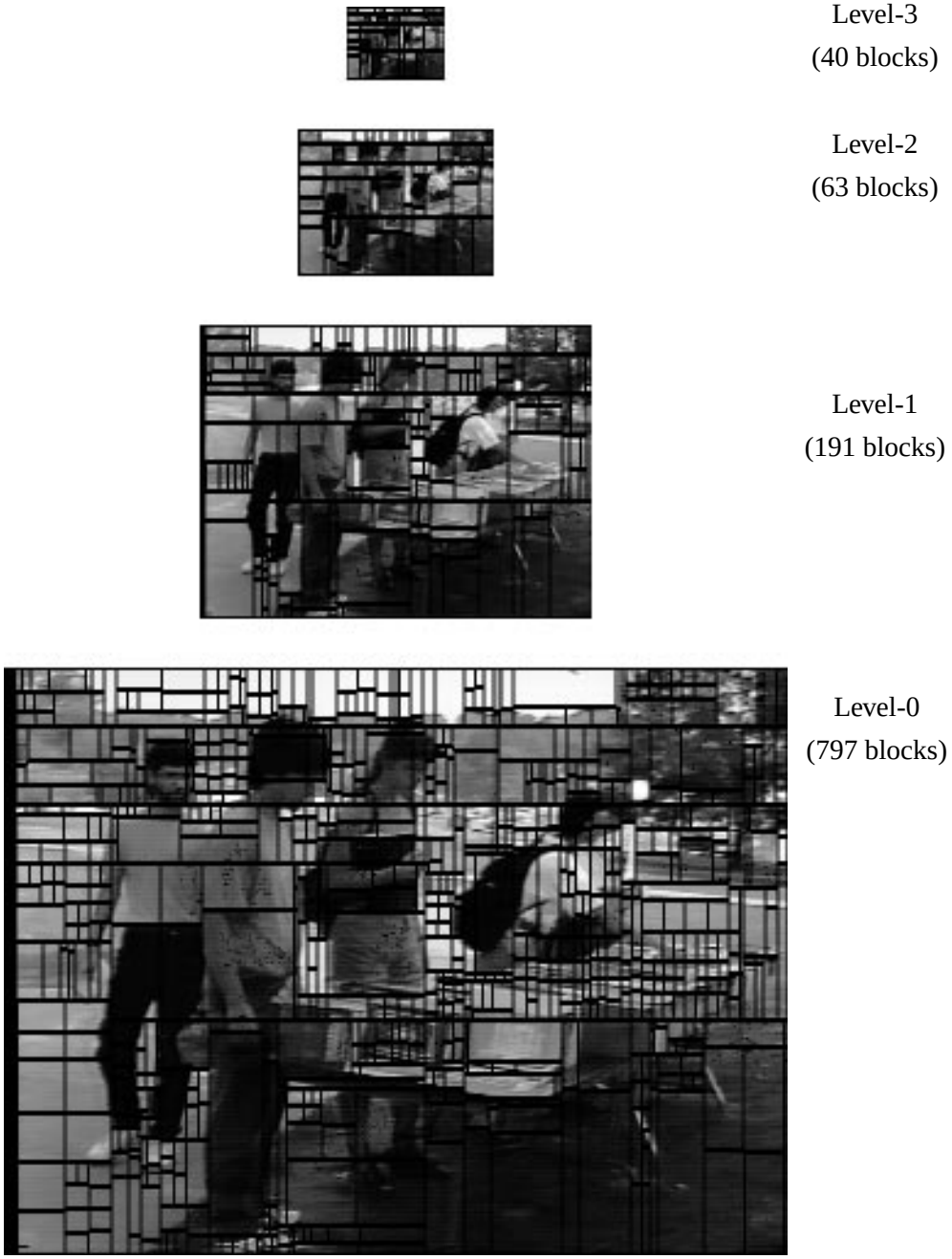


Fig. 3.6 (a) Multiresolution-based refinement of segmentation
(left image of a stereoscopic image pair from the booksale sequence)

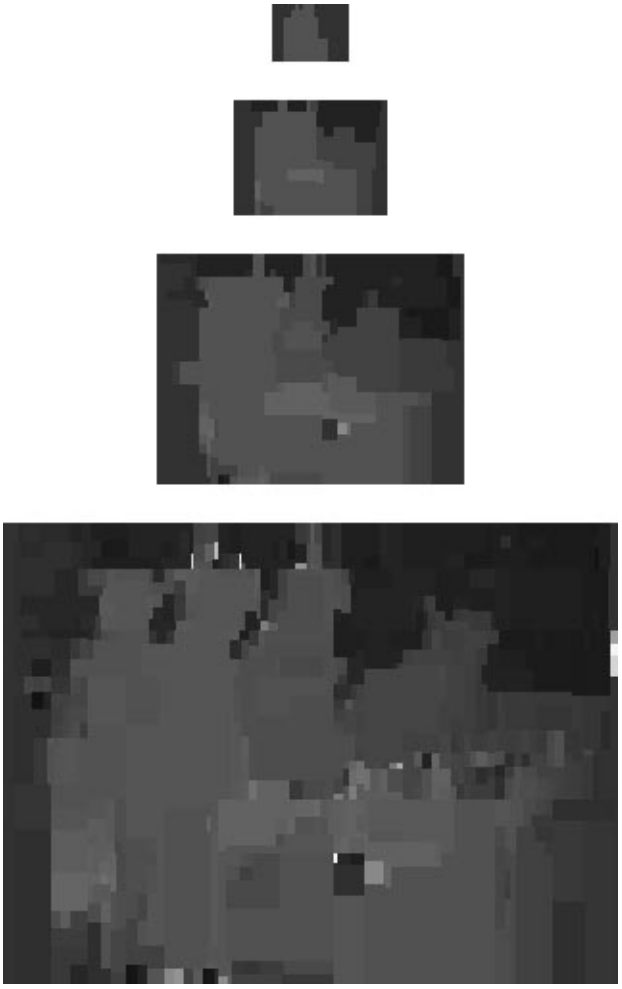


Fig. 3.6 (b) Multiresolution-based refinement of segment disparities
(left image of a stereoscopic image pair from the booksale sequence)

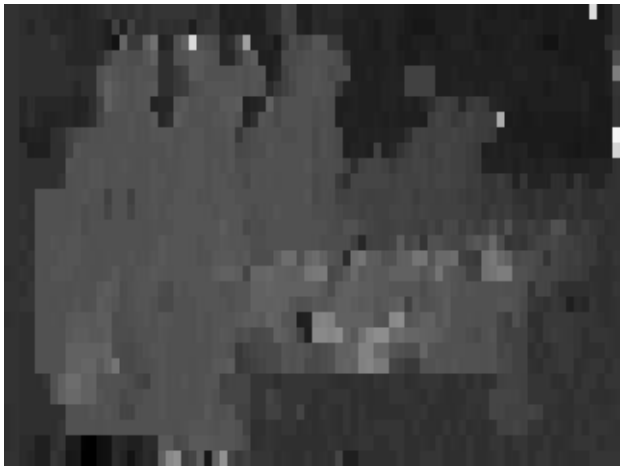


Fig. 3.6 (c) Disparity map obtained using FBS-based DCP with 8x8 blocks
The smoothness of the map is affected by the spurious matches.

Fig. 3.6 (d) Original left image

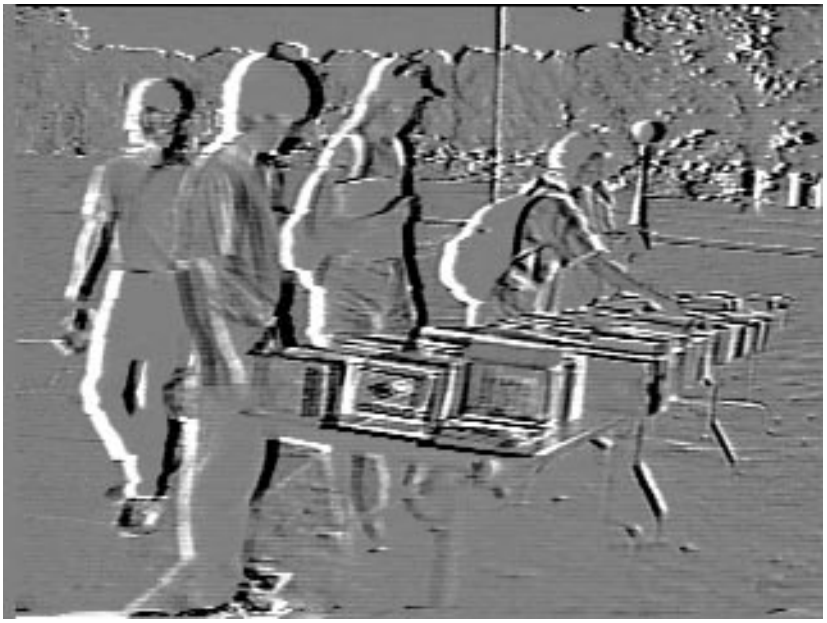


Fig. 3.6(e) Frame difference between the original left and right images (scaled by 2 and shifted by 128 gray levels)

Fig. 3.6 (f) Left image predicted from the right using DBS
(PSNR = 27.55 dB, bpp = 0.068)

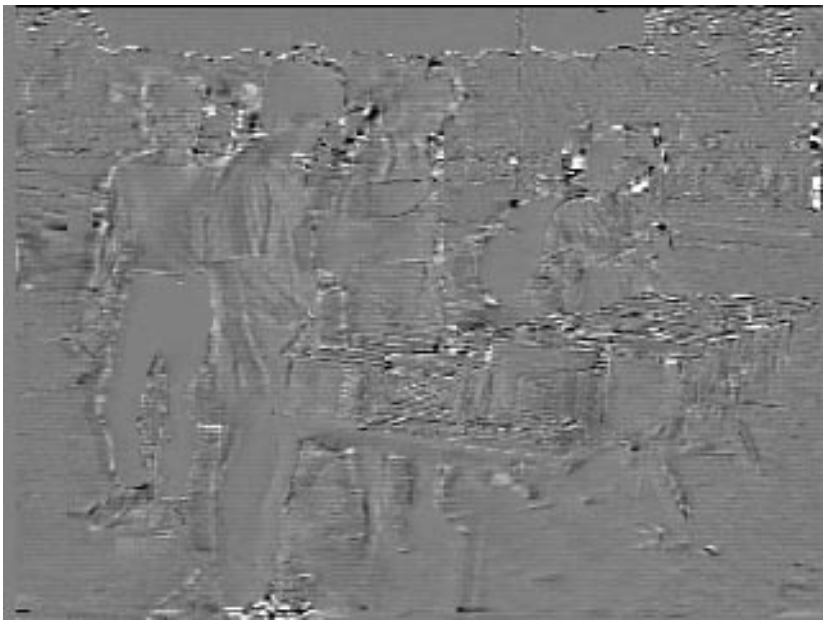


Fig. 3.6 (g) Prediction residuals (scaled by 2 and shifted by 128 gray levels)

FIG. 3.6: Results of DBS for a stereoscopic image pair from the booksale sequence

(a) Multiresolution-based refinement of segmentation (b) Multiresolution-based refinement of segment disparities (c) Disparity map estimated using FBS-based DCP (d) Original left image (e) Frame difference between the original left and right frames (f) Left image predicted from the right using DBS (g) Prediction residuals. (R_{dbs} = 0.6 R_{fbs}).

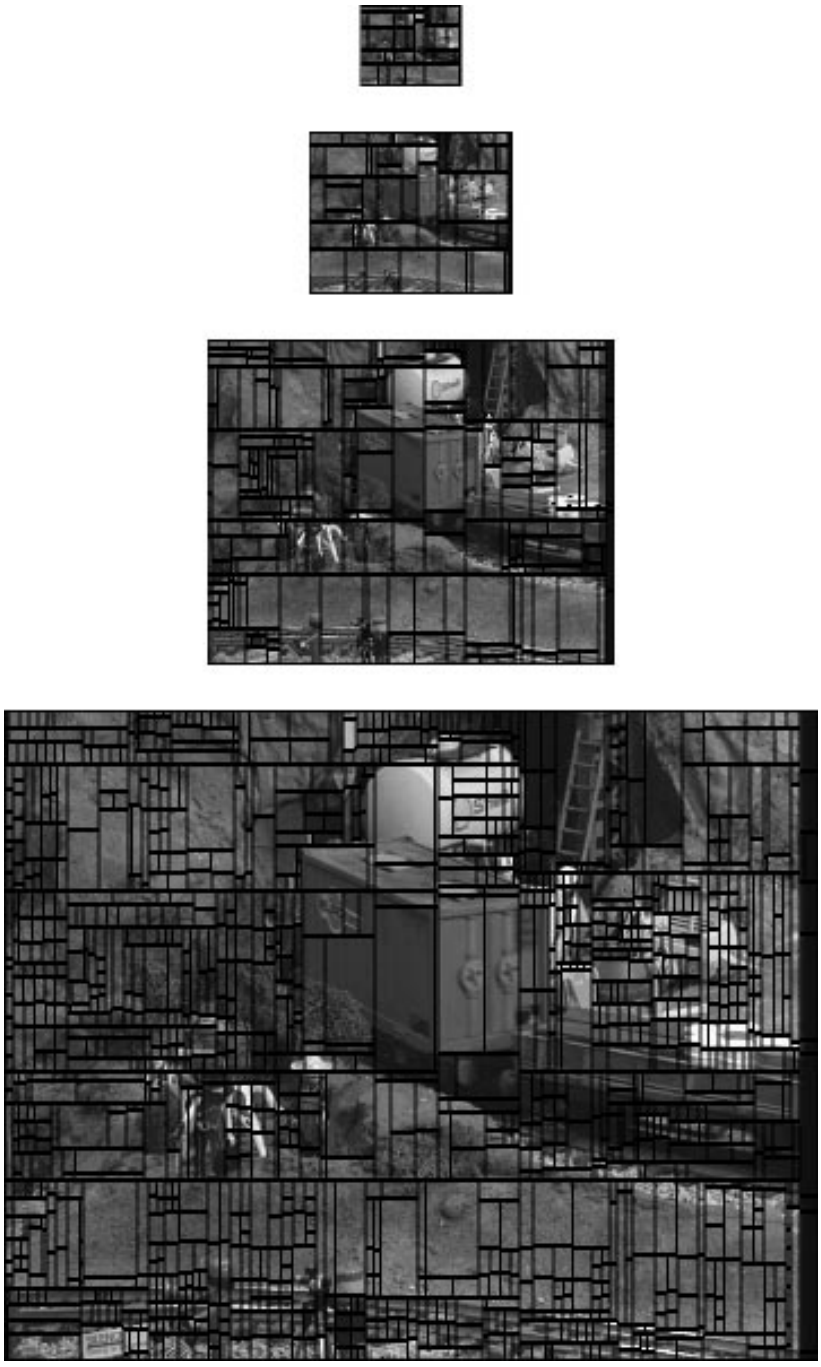


Fig. 3.7 (a) Multiresolution-based refinement of segmentation
(left image of a stereoscopic image pair from the *tunnel* sequence)

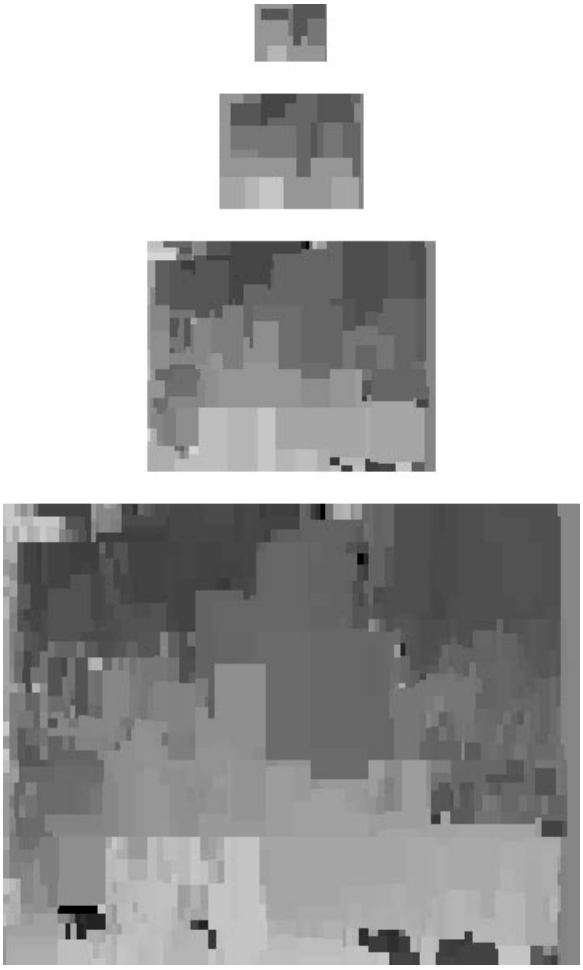


Fig. 3.7 (b) Multiresolution-based refinement of segment disparities
(left image of a stereoscopic image pair from the *tunnel* sequence)

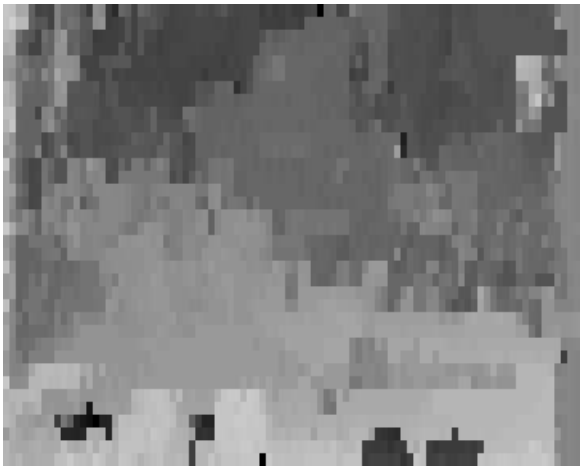


Fig. 3.7 (c) Disparity map obtained using FBS-based DCP with 8x8 blocks
The smoothness of the map is affected by the spurious matches.

Fig. 3.7 (d) Original left image

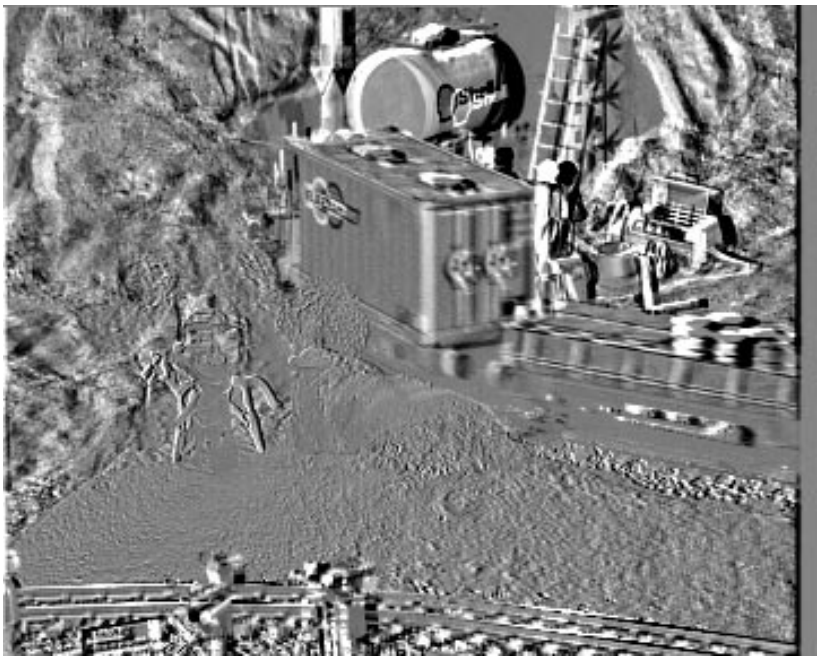


Fig. 3.7(e) Frame difference between the original left and right images
(Scaled by 2 and shifted by 133 gray levels)

Fig. 3.7 (f) Left image predicted from the right using DBS
(PSNR = 25.89 dB, bpp = 0.083)

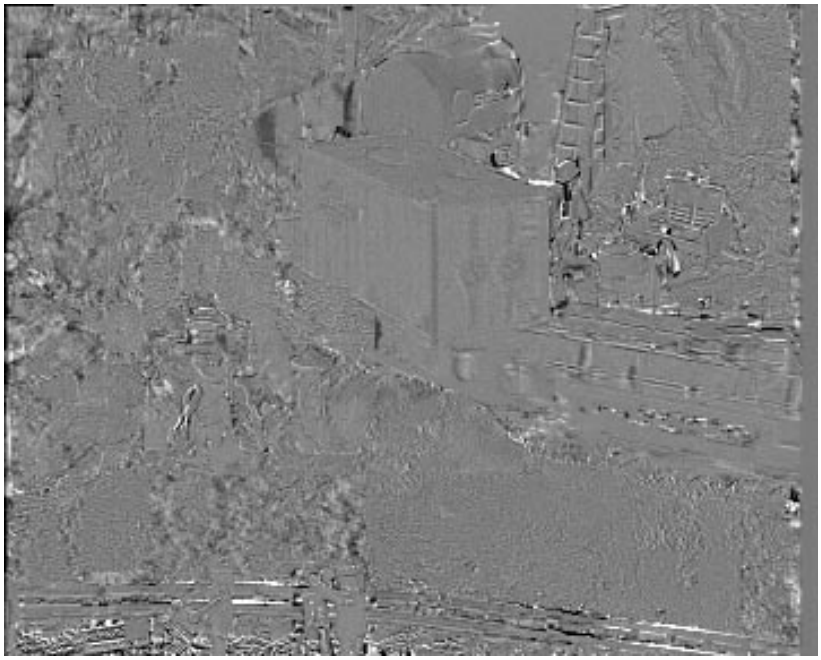


Fig. 3.7 (g) Prediction residuals (scaled by 2 and shifted by 133 gray levels)

FIG. 3.7: Results of DBS for a stereoscopic image pair from the *tunnel* sequence

(a) Multiresolution-based refinement of segmentation (b) Multiresolution-based refinement of segment disparities (c) Disparity map estimated using FBS-based DCP (d) Original left image (e) Frame difference between the original left and right images (f) Left image predicted from the right using DBS (g) Prediction residuals. ($R_{dbs} = 0.7R_{fbs}$).

an optimal partition in the R-D sense. This is because the distortion is not explicitly taken into account in the DBS algorithm in Section 3.2.5. In this section, we present a computational approach that will ensure that the ratio between the rate increase and the reduction in distortion due to a new partition stays within certain bounds.

Let λ be the worst case rate-to-distortion ratio that is permissible. To compute the rate increase due to partitioning, we need a model based on which the partitioning location and disparities are coded. The increase in the segmentation coding overhead when a node is chosen for splitting is modeled as follows:

$$\Delta R_{seg} = \log_2(S) + \left(\sum_{j=1}^{N} t_j \right), \quad \text{N} = 2 \text{ or } 4 \tag{EQ 3.6}$$

$$S = \begin{cases} K_v, & \text{for an arbitrary vertical split} \\ K_h, & \text{for an arbitrary horizontal split} \\ K_v + K_h, & \text{for a split in both directions} \end{cases}, \tag{EQ 3.7}$$

where K_v *and* K_h are the number of bits allocated for coding the partitioning location in the vertical and horizontal directions (Section 3.2.2), t_j is the tree structure coding overhead required for the *j*th child node which is given by,

$$t_j = \begin{cases} 1, \text{ if } (h_j < S_{min}) \text{ or } (w_j < S_{min}) \\ 2, \text{ otherwise} \end{cases} \tag{EQ 3.8}$$

The following model is considered for coding the disparities. The disparities estimated for the new child nodes are subtracted from the disparity of the parent node and these differences are coded based on their entropy over the entire image or a collection of images. Hence the additional bits needed to code the disparities of the child-nodes is given by,

$$\Delta R_{disp} = \sum_{j=1}^{N} d_j \quad , \quad N = 2 \text{ or } 4 \tag{EQ 3.9}$$

where d_j's are the lengths of the entropy-based codes for the *j*th child's disparity difference. Given these two models, the increase in the number of bits due to a new partition is,

$$\Delta R = \Delta R_{seg} + \Delta R_{disp} \tag{EQ 3.10}$$

The reduction in the distortion can be computed at each step from the distortion measures with and without the new partitions as,

$$\Delta D = MSE_{before} - MSE_{after} \quad \text{(EQ 3.11)}$$

In addition to the other splitting criteria, the following criterion can also be added to ensure that each split maintains the R-D below λ.

$$\frac{\Delta R}{\Delta D} < \lambda \quad \text{(EQ 3.12)}$$

Using this modification, several unnecessary partitions can be avoided. However, as the MSE evaluated at a lower resolution does not include the errors in the high frequency sub-bands, this procedure is applicable only at level-0. By specifying an appropriate λ for each resolution level, this procedure becomes applicable at all resolution levels.

3.2.8 Computational considerations

The computational complexity of the DBS algorithm depends on the disparity estimation as well as on the partitioning location computations. The disparity estimation complexity still remains nearly linear in terms of the total number of pixels per image (*N*). However, the constant attached is higher because disparity estimation for the same set of pixels is carried out once for each level of the quadtree above the leaf node containing those pixels. However, as a majority of these levels are at lower resolutions, the overall complexity when compared to a single-step hierarchical estimation is reasonable (say, a *factor of 3-4*). For instance, if only two levels of refinement on the average are required at the full-resolution and three levels on the average are required at the coarser resolutions, then from the search reduction analysis in Section 2.3.5, it can be seen that the complexity increases only by a factor of 3. The significant components of the partitioning location calculation are the computation of row and column means and their convolution with a symmetric difference filter. The complexity of each of these operations is linear in the number of pixels. Hence the partitioning location complexity is also *O(N)* and the associated constant is smaller than the constant in the case of disparity estimation.

Since the computing time at a given computational complexity can be decreased by parallelization, it is also important to explore the degree of parallelism offered by the DBS algorithm. Though each stage of the MR decomposition can be parallelized to a high degree, the subsequent stages of the decomposition depend on the outputs of the previous stages, thereby introducing a pipeline delay. During the segmentation, as we progress down the quadtree, the different nodes can be processed independently and hence a high degree of parallelism exists. However, this is lower than the parallelism available in a fixed block size case, as the degree of parallelism at the top of the quadtree is small. Also, owing to the different block sizes, each of the individual processing elements must be able to handle the largest possible block size. The S_{max} constraint on the maximum allowable block dimension makes this problem more manageable. The homogeneity based segmentation employed at the coarsest resolution level yields a fast initial

segmentation because of the reduced number of computations for such a segmentation[1] and the lower pixel count at the coarsest resolution. These segments from then on can be processed in parallel.

3.3 CONCLUSIONS

In this chapter, we have described and demonstrated a disparity-based segmentation approach for coding stereoscopic image pairs. The important contributions of this chapter are,

(1) the concept that segmenting a stereoscopic image pair based on the disparity detail present within the pair is optimal for DCP based coding, and

(2) the multiresolution based quadtree decomposition approach that simultaneously reduces the segment location coding overhead and the computational complexity.

It has been shown, through experimental results on a large set of stereoscopic image pairs, that DBS results in 25-55% savings in bits over FBS-DCP at similar quality levels. In addition, good compensation over a large percentage of the image is achieved with a very small number of blocks and thus the segmentation parameters can be varied to achieve graceful degradation in the quality after compensation, compared to the quality achieved by increasing the (fixed) block sizes. It also provides a disparity map more suited for intermediate view synthesis (see Chapter 5). The MR-based QTD can be used with other homogeneity criteria as well.

One limitation with the current implementation is that only horizontal and vertical partitions are allowed. Since the dominant edge within a block can have an arbitrary orientation, this restriction results in an increase in the leaf nodes. A variance-based segmentation technique that uses diagonal partitions in addition to the horizontal and vertical partitions is described in [41]. Since computations based only on the intensity values within each segment are needed in this case, a reasonable complexity is maintained. However, if we use such partitions for DCP, we will need to estimate disparity over triangles and irregular quadrilaterals. This significantly increases the computational complexity during block matching, as each pixel position would have to compared with the equation of one or more lines to determine whether it belongs to a candidate search block.

Though the splitting criteria in step 6 of the DBS algorithm in Section 3.2.5 ensures that a node is split only if its child nodes have different disparities, there can be spatially adjacent leaf nodes that share the same disparity value. The motivation of split and merge methods [73] is to merge these regions to obtain homogeneous (but arbitrarily shaped) areas, which can be used in applications such as pattern recognition. However, from the coding point of view, a label has to be assigned to each group of spatially adjacent leaf nodes that share the same depth. In a general

1.The squared values of the pixel intensities needed for the variance computation need to be computed only once per pixel.

image, the number of identifiable physical objects at different depths is usually larger than the number of disparity levels (assuming half-pixel accurate disparities and NTSC resolution). Hence, coding the label for each block and coding the disparity of each label requires more bits than just coding the disparity of each block. In addition, the differential predictive coding of the node disparities on the quadtree exploits the similarity in the disparity of spatially adjacent blocks. For this reason, we have not considered a merging step in our algorithm.

In this chapter, we considered the compression of still stereoscopic image-pairs. In the next chapter, we consider the compression of stereoscopic image sequences, where we apply the MR-based QTD approach to achieve motion-adaptive segmentation as well. In Chapters 4 and 5, we present some modifications to the segmentation that would make the coding rate and complexity scalable with multiple views.

Chapter 4

Stereoscopic sequence compression

In Chapter 3, we considered the problem of compressing still stereoscopic image pairs, and demonstrated that the disparity-adaptive segmentation using the multiresolution-based quadtree-decomposition (MR-QTD) method results in a considerable increase in coding efficiency for disparity compensated prediction. In this chapter, in addition to extending the above segmentation technique to fit within a sequence coding framework, we address several critical issues affecting stereoscopic sequence compression (SSC) and propose solutions for,

(1) exploiting intra-view and inter-view redundancies to increase coding efficiency,

(2) adapting the excess bandwidth needed to transmit stereoscopic video to be commensurate with the demand for stereoscopic video,

(3) exploiting properties of human visual system specific to stereoscopic perception, and

(4) joint coding of the sequences to improve the scalability of computational and coding efficiencies with multiple views.

We consider compression of 2-view sequences in this chapter. Possible multi-view extensions are presented in Chapter 5. However, we allude to scalability of computational complexity or compression efficiency with multi-views where appropriate. In Section 4.1, we introduce the problem of stereoscopic sequence compression. Different frame prediction possibilities in a stereoscopic sequence are discussed in Section 4.1.1. The different factors that influence the choice of prediction modes are identified in Section 4.1.2. Based on these observations, we describe two basic configurations for SSC in Section 4.1.3. A residual coder that is suited for low bit-rate coding is developed in Section 4.2. In Section 4.3, we present a FBS-based baseline scheme for each of the configurations, against which the segmentation-based SSC extensions can be compared. Two dependent coding extensions that use MR-QTD are developed in Section 4.4. Two segment-tracking based joint-coding extensions to improve the computational and coding performances are introduced in Section 4.5. A mixed-resolution based coding approach that readily fits into the multiresolution framework is presented in Section 4.6, along with the psychophysical motivation behind such coding. Rate-distortion performances over a test set of stereoscopic image sequences are presented for the different SSC extensions in Section 4.7, along with the inferences. A summary of the different extensions introduced in this chapter along with their salient features is presented along with the conclusions in Section 4.8.

4.1 INTRODUCTION

Typical image sequence compression methods were described in Section 2.2.5. These methods exploit the spatial redundancy within a frame, the temporal redundancy between adjacent frames, and tolerances of the human visual system to achieve very high compression ratios. The simplest conceivable stereoscopic sequence compression method would be to independently code each of the views using such sequence compression methods. In this case, an *n*-view sequence would require *n* times the bit-rate needed to transmit a single sequence. To achieve any significant bandwidth reduction, compared to such independent coding, we need to consider several additional factors such as, cross-stream correlation and psychophysical factors associated with stereoscopic perception. The problem is made more difficult due to other practical considerations, such as the need for a disparity map at the decoder to synthesize intermediate views (see Section 2.1.5), unavailability of excess bandwidth, moderate encoder and low decoder complexity requirements, and the need for quality compatibility with existing monoscopic transmission schemes. In the following subsection, we present a frame structure for coding stereoscopic sequences that would enable us to exploit cross-stream correlation while retaining some of the desirable features of conventional monoscopic sequence compression methods.

4.1.1 Frame structure for stereoscopic sequence compression

The frame structure recommended by the MPEG video coding standards (Section 2.2.7) has several attractive features. The independently intra-coded I-frames enable random access, editability, and independent decodability of different segments of a coded sequence. The I- and P-frames serve as periodic reference from which the intermediate B-frames are predicted. To prevent the accumulation of prediction errors over time due to prediction from progressively lower quality frames, the I and P-frames are typically coded at a higher quality than the B-frames. The coding efficiency for B-frames is improved by employing bi-directional prediction, albeit at the expense of increased computational burden, as regions occluded in one reference frame can be predicted from the other reference frame. To retain these features while coding a stereoscopic sequence, we consider a similar frame structure in this thesis.

Compared to independent coding of the multiview sequences, additional compression can be achieved by exploiting the intra-view inter-view redundancies that exist. Let us assume that one of the sequences is coded independently, while the other sequences are coded w.r.t this independently coded sequence. Such a coding scenario is referred to as *dependent coding*. Assuming an MPEG-like frame structure in each of the views, the I-frames of these other views can be predicted using disparity compensation w.r.t the I-frame of the independently coded sequence[1]. Since, intracoding

1. Of course, ‘I-frame’ would be a misnomer for these frames. The notation can be considered to represent frames that correspond to the I-frame in the independently coded sequence.

typically accounts for 20-30% of the overall bit-rate, the most significant reduction in bit-rate would come from this step. In addition, the P-frames in these views can be bidirectionally predicted w.r.t a past reference frame within that view and w.r.t the corresponding frame in the independently coded sequence. Since the correlation with the corresponding frame in the other view is likely to be higher than the correlation with the previous reference frame within a view (for a sequence with moderate camera and object motions and for a typical P-to-P frame separation), this step would also contribute to a reduction in bit-rate. The reduction in bit-rate can also be attributed to the fact that a region occluded in the temporal reference frame can be predicted from the corresponding view (provided it is not occluded in perspective as well). Similarly, the B-frames can be tri-directionally predicted. These prediction modes are illustrated in Fig. 4.1. Bidirectional prediction w.r.t a temporal frame and a corresponding frame from the other view has been discussed in [34 and 96].

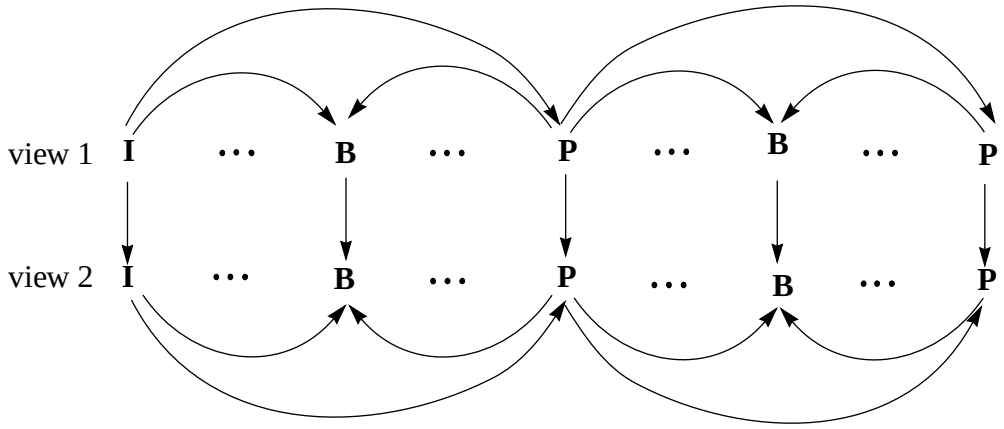


FIG. 4.1: Dependent coding - prediction modes for the different frames
(An MPEG-like frame structure is assumed)

4.1.2 Factors influencing the prediction modes

In the above discussion, we did not consider the quality of the reference frames specifically. However, as we mentioned in Section 1.3, the demand for stereoscopic video may never be high enough to warrant an *n*-fold or nearly *n*-fold increase in bandwidth in a broadcast-type application. Since most viewers are likely to watch monoscopically at any given time, at least one sequence within the multi-view sequences should be coded at a higher quality. We refer to such a sequence as the *main sequence*. The other sequences that are coded at a quality commensurate with the demand for stereoscopic video and the functional advantages that stereoscopic video offers, will be referred to as *auxiliary sequences*. We denote the frames in the auxiliary sequences that correspond to the I-, P-, and B-frames of the main sequence by I_A-, P_A- and B_A-frames respectively. In this chapter, we restrict our attention to one auxiliary sequence. In Chapter 5, we consider the coding of multiple auxiliary sequences.

The difference in quality levels, between the different frames within a sequence and across views, has a considerable influence on the particular prediction mode that would be favored during SSC. For instance, if the auxiliary sequence is coded at a significantly lower quality than the main sequence, then DCP would be favored over within-the-sequence MCP for the P_A- and B_A-frames. Similarly, as the B-frames in a sequence are coded at a lower quality than the I- and P-frames, if the auxiliary sequence is coded at a rate similar to that of the main sequence, then MCP would be favored over DCP for the B_A-frames. Though reduced quality coding of auxiliary frames has been considered in the literature [96], the excess bandwidth is arbitrarily chosen. Also the impact of the reference frame quality has not been addressed by other researchers.

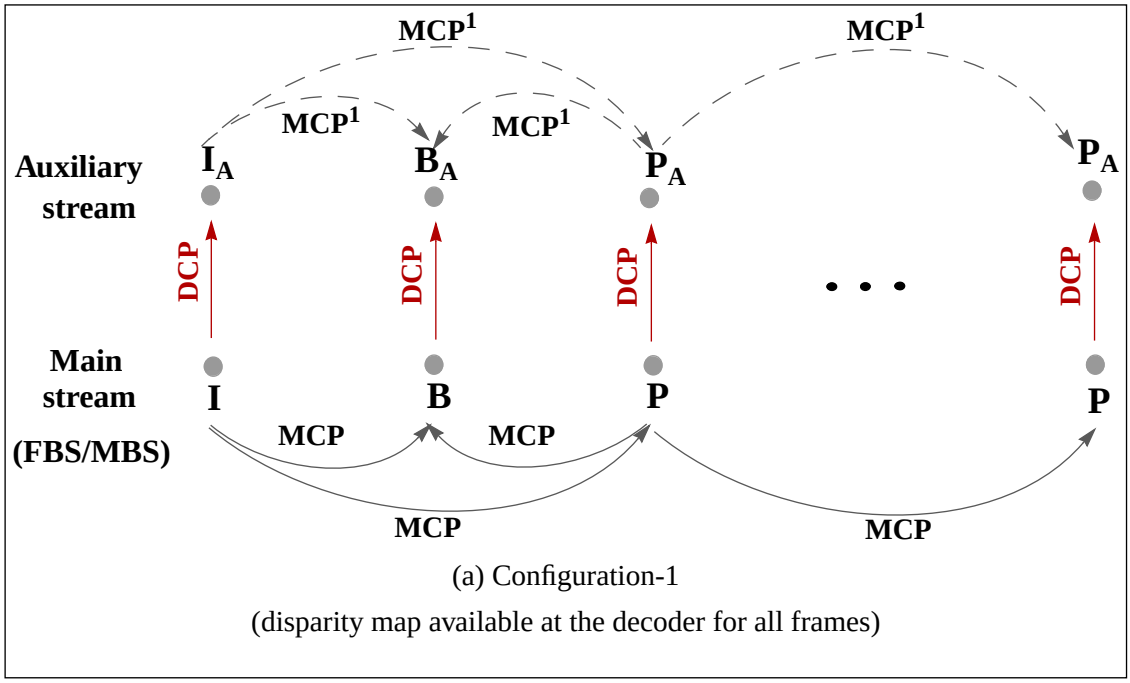
The choice between DCP and MCP for coding an auxiliary frame also depends on the following factors. (1) inter-frame motion (extent, rotational components and scale changes), (2) extent of disparity or the distance of objects from the cameras, (3) disparity being a scalar (as opposed to being a two-component vector like motion) for a parallel axes imaging geometry, (4) the match between the left and right cameras in terms of brightness, contrast, and color balance, and (5) the need for synthesizing intermediate views at the decoder.

4.1.3 Configurations for stereoscopic sequence compression

While most of the above-mentioned factors influence the choice of DCP vs. MCP at a per-block basis, the need for a complete disparity map at the decoder precludes the possibility of using MCP altogether. For this reason, we consider two basic configurations, configuration-1 and configuration-2, for coding the auxiliary stream. In configuration-1, the auxiliary sequence frames are estimated through DCP. Hence a complete disparity map will be available at the decoder for each frame. Under-compensated regions due to occlusions and DCP errors are further compensated through MCP w.r.t past and future reference frames in the auxiliary sequence. In configuration-2, the auxiliary frames are estimated through bidirectional prediction w.r.t the corresponding main sequence frame and the nearest reference frame in the auxiliary sequence. This configuration thus has the capability to choose adaptively between DCP and MCP. However, the decoder no longer has the complete disparity map and hence synthesis of intermediate views is not possible. The second configuration is similar to the configuration-2 described in [96], in which an MPEG-2 compatible stereoscopic sequence coding scheme by using MPEG-2's temporal scalability model is considered. Our two basic configurations are illustrated in Fig. 4.2.

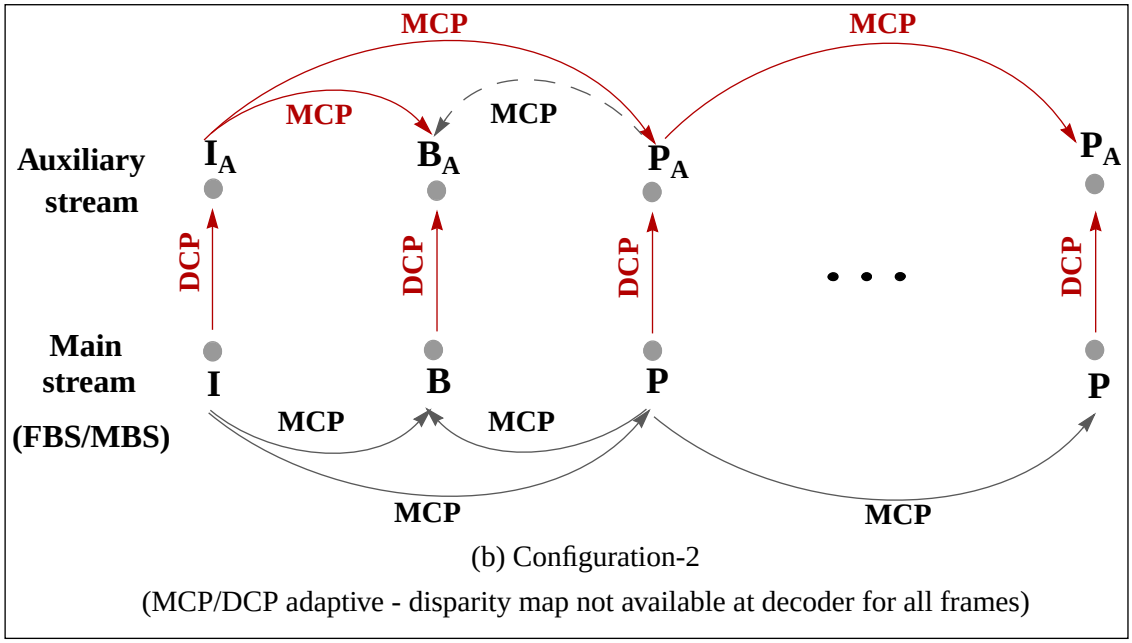
4.2 RESIDUAL CODER

Before considering the SSC schemes, we briefly describe the residual coder that we employ. Though motion or disparity compensated prediction typically provides acceptable compensation for most regions in an image frame, significant errors can be present at least in a few regions owing to the failure of the assumptions behind block based compensation e.g., failure of translational



(a) Configuration-1

(disparity map available at the decoder for all frames)



(b) Configuration-2

(MCP/DCP adaptive - disparity map not available at decoder for all frames)

I - Intracoded frame P - Predicted frame B - Bidirectionally predicted frame

I_A,P_A,B_A - corresponding frames in the auxiliary stream

MCP - Motion compensated prediction DCP - Disparity compensated prediction

MCP^1 - MCP applied only when the block is under-compensated after DCP

FIG. 4.2: Stereoscopic sequence compression - two basic configurations

displacement or constant disparity over a block assumption, and partial occlusion of a block. Significant residuals, if left uncoded, can result in severe degradation in the perceived quality of an image, and due to the inter-frame predictions, the errors will also accumulate over time. However, because of the high entropy of the residuals, even their lossy coding typically constitutes a significant fraction of the overall bit budget. The MPEG standard recommends a discrete cosine transform based residual coder [25]. However, the residuals contain no special structure in the transform domain which can be exploited to code them efficiently. In fact, if the residuals are sparse within a block (which is more likely), the number of significant non-zero values in the transform domain will be higher than the number of significant residuals in the spatial domain. Due to the reduced bit-budget for the auxiliary sequence, we need a residual coder that can target bits specifically to regions with significant errors, so that the most distracting errors can be coded within a limited bit-budget.

Selective residual coding requires coding the locations of significant residuals in addition to coding the values of the residuals. To capitalize on the sparse nature of the residuals, we adopt a quadtree-based residual coder to reduce the overhead associated with coding the locations. A vector quantizer / scalar quantizer combination is used to code the error values at the different block sizes of the quadtree. Each residual frame is divided into blocks of size 16x16 referred to as macroblocks (as in the MPEG standards). Two measures of distortion are used in deciding whether a block needs to be coded or not. One is the *MAE* defined as,

$$\sum_{k \in \eta} \left| I_{act}(k) - I_{est}(k) \right| \qquad \text{(EQ 4.1)}$$

where I_{act} is the actual image, I_{est} is the estimated image and η is the set of all pixels in the block. The other is the *significant error count* (N_T) defined as the number of pixels for which,

$$\left| I_{act}(k) - I_{est}(k) \right| > T, (k \in \eta) \qquad \text{(EQ 4.2)}$$

where T is some pre-specified significant error. Two thresholds, namely the maximum allowable *MAE* (E_{max}) and the maximum allowable significant error count (N_{max}), typically 0 or 1, are specified for each frame. If ($MAE > E_{max}$) or ($N_T > N_{max}$) for a block, then that block is considered for residual coding.

The macroblock size is chosen as 16x16 to keep the depth of the quadtree small and to enable a certain degree of parallelism. Also, for typical images, a larger block size has a higher likelihood of containing significant errors. The quadtree based VQ/SQ residual coding algorithm for each MB is summarized in Table 4.1.

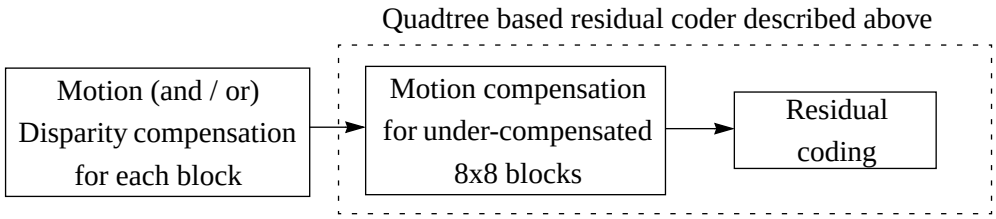
The codebooks are generated using the LBG algorithm [43]. The 16-dimensional vector codebook is obtained by training over a set of residual code vectors derived from typical sequences. A subset of training vectors with errors in the range (-32,32) gray levels is chosen for

TABLE 4.1: Summary of the quadtree and VQ/SQ based residual coder

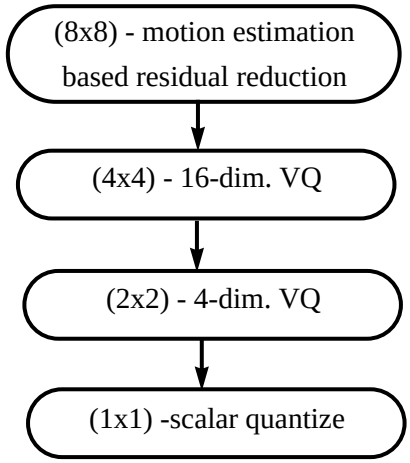
Block Size	Step 1: If ($MAE > E_{max}$) or ($N_T > N_{max}$)	Step 2: If ($MAE > E_{max}$) or ($N_T > N_{max}$)
16 x 16	Divide into four 8x8 subblocks	
8 x 8	Perform MCP w.r.t to a reference frame, if necessary. If the resulting MAE is less than 70% percent of the previous *MAE*, code the motion vector. Compute N_T.	Divide into four 4x4 subblocks.
4 x 4	Compute mean squared errors (*MSE)* w.r.t the codevectors of a 16-dim. vector codebook. Pick the codevector that produces the least *MSE*. Compute new *MAE* and N_T.	Divide into four 2x2 subblocks. Else, code the VLC corresponding to the best matching codevector.
2 x 2	Compute *MSE* w.r.t the codevectors of a 4-dim. vector codebook. Pick the codevector that produces the least *MSE*. Compute new *MAE* and N_T.	Divide into four single pixels. Else, code the VLC corresponding to the best matching codevector.
1 x 1	Compute the nearest quantizer level in a scalar quantizer. Code the VLC corresponding to that level.	

actual training, and vectors with larger errors are relegated to the subsequent levels of the quadtree. The entropy of each codevector over the training set is used to assign a variable length code (VLC) to that codevector. The 4-dimensional vector codebook is obtained in a similar manner with a larger range for the residuals. The scalar quantizer levels are designed for the Laplacian distribution of the errors [44] obtained from actual runs of the residual coder incorporating the above two VQs. The quadtree structure coding overhead (1 bit per node) and the VLCs from the VQ and SQ stages constitute the residual coding overhead for a macroblock.

Control of this residual coder is achieved via two quality metrics, namely, *MAE* and N_T (for a specified T). These metrics only ensure constant quality; precise rate control is not possible. However, it is possible to get close to a desired bit-rate by adaptively setting the thresholds in the coder based on the knowledge of the bit rates of previously coded frames. The thresholds for the quality metrics can be augmented with a threshold for the ratio between the error variance within a block and the intensity variance (or spatial activity) within that block. Compared to thresholding based on the error statistics alone, such a threshold exploits the masking effects inherent in the human visual system to target the residual coding bits accordingly. For instance, a particular error



(a) Stages involved in coding a typical non-intra coded frame



(b) Residual coding methods at the different levels of a macroblock quadtree

FIG. 4.3: Quadtree and VQ/SQ based residual coder

variance that is acceptable in a block with a high spatial activity can be objectionable in homogeneous blocks.

4.3 BASELINE SCHEMES

We introduce initially two baseline SSC schemes, one for each configuration. The baseline schemes employ fixed block-size based disparity and motion compensation (in Fig. 4.2 (a) and (b)) and are representative of the MPEG standards. These baseline schemes, referred to as FBS-1 and FBS-2 schemes to denote the use of fixed block sizes and the coding configurations, are used to outline the details behind SSC; they also serve as references against which the MR-QTD based extensions that are presented in the later sections will be compared.

The sequence inputs to the compression scheme are in YUV 4:2:0 format (see Section 2.2.7 on page 19). The main sequence is coded independently through FBS based MCP with an MPEG-like frame structure. The Y, U, and V components of the I-frames are coded using DCT-based intracoding of 8x8 blocks, described in Section 2.2.3 and shown in Fig. 2.4. The Huffman tables from the MPEG-2 recommendations are used to runlength code the quantized DCT coefficients

after the zig-zag scan [25]. Hierarchical block matching as described in Section 2.3.5 and Fig. 2.9 is employed for MCP and DCP. Since it is hard to achieve a perfectly parallel camera configuration, a small search range is allowed in the vertical direction during disparity compensation. The predictive models used for coding the motion and disparity vectors are described in Appendix-B. The residuals are coded by setting the E_{max} and N_{max} threshold parameters for the residual coder (described in the last section).

4.4 MR-QTD BASED DEPENDENT CODING EXTENSIONS

In this section, we consider two simpler extensions that incorporate the multiresolution-based quadtree decomposition approach within the two baseline configurations.

4.4.1 Extension-1 (DBS-1)

The FBS-1 baseline scheme can be extended in a straightforward manner to incorporate the MR-QTD approach by replacing the fixed-block-size based disparity compensation with the DBS algorithm developed in the last chapter. We refer to this extension as DBS-1. All the results that applied for a single frame coded using DBS in the last chapter would apply to coding the auxiliary sequence frames.

4.4.2 Extension-2 (DBS-2)

The DBS algorithm is applicable only for disparity compensated prediction. Different parts of an object at a particular depth (from the camera) can undergo different displacements over time - e.g., an object rotating about an axis parallel to the camera axis. The FBS-2 scheme involves a bidirectional prediction using motion and disparity compensations. To incorporate the MR-QTD method within this scheme, the DBS algorithm has to be extended to include motion based segmentation as well. This is done by estimating both motion (w.r.t a reference frame in the auxiliary sequence) and disparity for each segment in the DBS algorithm in Section 3.2.5. The partitioning criteria in step 6(c) of the algorithm are modified as follows:

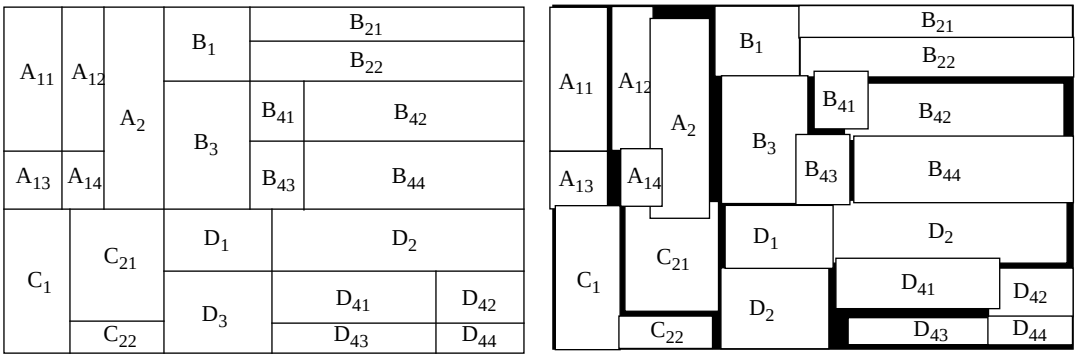
> If ((the difference between the sub-block disparities > D_{max}) and (the difference between the sub-block displacement components > M_{max})) or ($h > S_{max}$) or ($w > S_{max}$), then divide the block at the predetermined locations (where M_{max} is the maximum allowable absolute difference in a displacement component between sub-blocks).

Since a good match is needed only in either one of the reference frames, a block is split only if the sub-block motions as well as the sub-block disparities are different. Hence this segmentation typically results in fewer segments than DBS-1. We refer to this SSC extension as DBS-2.

4.5 MR-QTD BASED JOINT CODING EXTENSIONS

4.5.1 Reversal of prediction direction

Segmentation using MR-QTD requires a coding overhead as well as a computational overhead. The DBS-1 and DBS-2 extensions require each frame to be segmented. Also, the main sequence in these extensions are independently coded using a FBS-based MCP. This sequence can also be coded using motion-adaptive segmentation. Such additional segmentation would increase the computational burden further. Hence it would be preferable if the same segmentation could be used to code several frames, either along the view dimension or along the temporal dimension, so that the computational overhead and the segmentation coding overhead can be shared by all these frames. However, the quadtree-based representation is a nested spatial representation and cannot be used when its leaf nodes are undergoing independent spatial displacements. This precludes the possibility of using the same quadtree representation for all the frames while performing motion or disparity estimation in the *forward* direction. Sharing the segmentation overheads, then, requires a *reversal in the direction of prediction*. In other words, the segments in one frame can be *tracked* to other frames. This constitutes a significant shift in the paradigm compared to the conventional estimation. In conventional estimation, the frame to be coded is partitioned into non-overlapping blocks and the best match for each of these blocks is searched in the reference frames. In this case, some reasonable prediction (not necessarily meaningful) is obtained for all the blocks. However, the reversal of prediction direction results in a predicted frame with some regions that have no prediction (*holes*) and some regions that have multiple predictions. This is illustrated using Fig. 4.4.



Segmented frame - A (non-overlapping partition)

Frame predicted by tracking the blocks - B

Corresponding segments in B overlap due to motion occlusions & motion estimation errors

New regions (shown in black) that were occluded in A are exposed in B

FIG. 4.4: Impact of the reversal of prediction direction

As objects within the scene undergo displacement, new regions may be exposed and currently exposed regions may get occluded. If a segment in frame-A is occluded partially in frame-B (in

Fig. 4.4), then the best match for that segment can occur at the correct location, or a spurious match may be generated, depending on the extent of occlusion and the chance existence of other good matches. When the match occurs at the correct location, the occluded region has two candidate matches - one corresponding to the occluded region and the other corresponding to the occluding region. For example, in frame-B, a portion of segment B_{41} occludes a portion of segment B_{22}. The common region between these two segments thus has two possible candidate matches. When a spurious match occurs, the corresponding segment leaves behind an unfilled region and also adds itself as a candidate estimate for the false match location. The regions uncovered while tracking the segments, by definition, have no predictions.

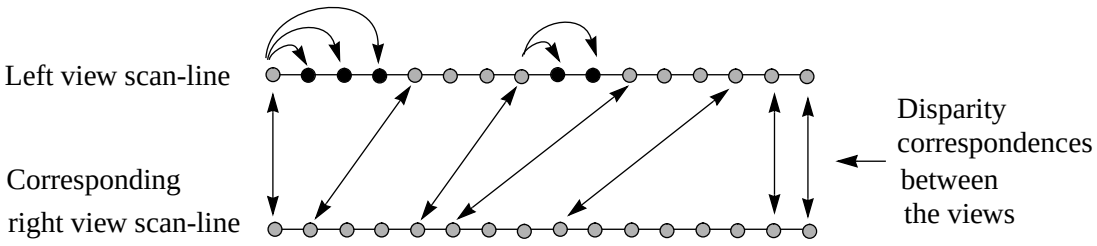
To code the frame under consideration, we need to (a) choose the correct match among the multiple matches and (b) obtain suitable predictions for the exposed regions. When reliable disparity estimates for the segments in frame-A are available, we can use the depth-order provided by these disparity estimates (i.e. the fact that a segment that is farther away cannot occlude another segment that is closer to the camera) to disambiguate between multiple matches. The regions without prediction can be intra-coded. But due to the arbitrary location and irregular shapes of these regions, the intra-coding overhead would be high. Interpolation based filling-in of these regions can result in loss of quality.

4.5.2 RDBS scheme

In this section, we consider a joint coding scheme in which the main sequence is also coded using motion-adaptive segmentation. Each frame in the main sequence is segmented using the DBS algorithm. Thus the main stream frames have non-overlapping partitions. Motion compensation for the P- and B-frames are carried out over these variable-size blocks. To account for independent sub-block displacements within a block, these blocks are further partitioned with the error after motion compensation as the splitting criterion. The disparity map, computed during the segmentation, is *reversed* to predict the auxiliary sequence frames from the main stream frames. Thus, each stereoscopic pair of frames share the segmentation coding overhead. The non-overlapping partitions in the main stream frame overlap in the predicted auxiliary sequence frame and *holes* arise at locations that correspond to occluded regions and regions with disparity estimation errors. Multiple candidate matches during the reversal are disambiguated using the disparity. However, the cost of coding the holes (regions where no prediction is available) can partially offset the gain in bit-rate achieved through joint coding. We introduce a method for filling-in these uncovered regions in the following subsection.

Spatial prediction for uncovered regions

Given that the stereoscopic pair of frames are captured at the same time, the disparity map depends only on the depths of the different objects at that instant and the fixed binocular camera geometry. Thus, occlusions due to binocular parallax are more structured than motion-based



Dark pixels, the pixels without prediction in the left view scan-line, are predicted from the nearest background pixel

(a)

(b)

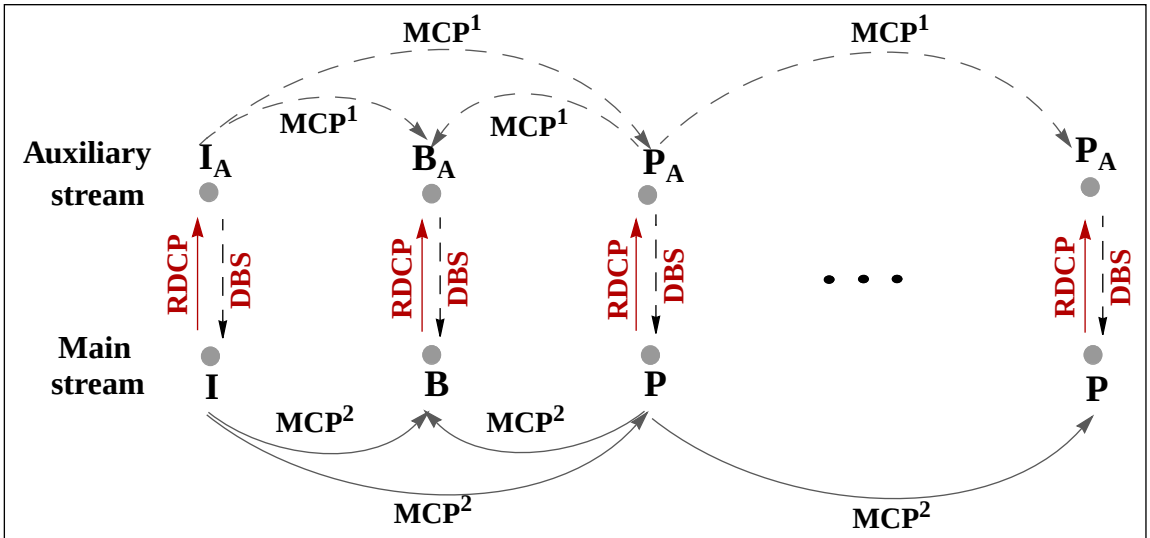
(c)

FIG. 4.5: Spatial prediction for regions uncovered during reversal of prediction direction
(a) A simple illustration (b) A D_{BR} frame estimated by reversal of disparity map (regions without estimate appear as black patches) (c) Frame in (b) after the described concealment

occlusions which depend on the displacements of the different objects in the scene. A scan-line algorithm for filling the holes can be developed, given that the camera axes are parallel. By assuming that an uncovered region is part of an object that is at a greater depth than the object that exposed that region, a spatial prediction for the uncovered region can be formulated. Operating along scan-lines and using the estimated disparity map, the direction (left or right) of the background object near an exposed region can be found. The intensity value of the neighboring background pixel is made the prediction for all the exposed pixels on a scan-line. Such a unidirectional prediction ensures that an erroneous interpolation is not carried out across two regions with different disparities. For typical scenes, the filled-in value is close to the actual intensity value for most pixels in most of the holes. No coding overhead is incurred for such a prediction scheme. However, as the decoder also has to perform the detection and prediction of holes, its complexity is increased. This scheme is illustrated in Fig. 4.5(a); the effectiveness of the method is shown for an auxiliary frame from the *booksale* sequence in (b) and (c).

A half-pixel accurate motion estimation is carried out for each tracked block in the reference frames. Unlike the typical single-directional estimation where the half-pixel accuracy can be coded using one additional bit for each direction, in this case, we need two bits per direction to code the three possibilities of -1/2, 0, and +1/2 pixel displacements[1]. After filling-in the exposed

regions, the residuals are coded using the coder in Section 4.2. Since the auxiliary stream frames are obtained by reversing the direction of prediction, we refer to this scheme as RDBS (reversed DBS). The different prediction modes are illustrated in Fig. 4.6. This scheme belongs to configuration-1 as the decoder has a complete disparity map for each frame.



DBS - Main stream frames are segmented using DBS

RDCP - Aux. frame predicted by *reversing* the DCP direction and then filling-in the *holes*.

MCP^1 - MCP w.r.t to reference frames (only for under-compensated blocks).

MCP^2 - MCP of the segments - under-compensated segments are further segmented

FIG. 4.6: RDBS scheme - configuration 1

4.5.3 Segment Tracking (ST-1)

In the RDBS scheme, the segmentation has to be repeated for every stereoscopic pair of frames. The computational and coding overheads associated with segmentation can be further reduced if a group of stereoscopic pairs of frames share the same segmentation. This can be achieved by segmenting a reference frame and *tracking* the segments in both the streams up to the next reference frame. Since the segmentation has to hold for both disparity and motion compensation, again a joint motion and disparity based segmentation (MDBS) is required as was described in Section 4.4.2, but with the following modification. The motion-adaptive segmentation is performed w.r.t the nearest future reference frame within the sequence, and the criteria for dividing a block (step 6(c) of Section 3.2.5) is:

> If ((the difference between the sub-block disparities > D_{max}) or (the difference between the sub-block displacement components > M_{max})) or ($h > S_{max}$) or ($w > S_{max}$), then divide the

1. By entropy coding or by coding the half-pixel estimates over several blocks, the average number of bits needed per estimate can be made arbitrarily close to $\log_2(3)$ bits.

block at the predetermined locations (where M_{max} is the maximum allowable absolute difference in a displacement component between sub-blocks).

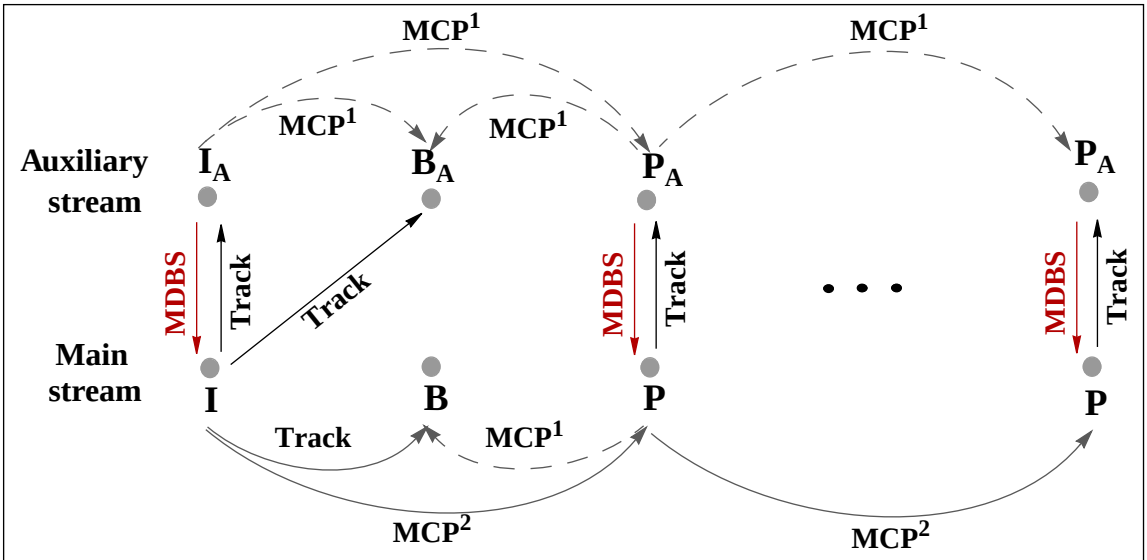
Such a segmentation typically results in more segments than the DBS algorithm, as good compensation in both the temporal and perspective domains is required.

The main stream's reference frames are segmented using MDBS. The main stream's B-frames are estimated by tracking the reference frame's segments and then reversing the direction of prediction. Since the same segment is tracked, the motion estimates from the segmentation with proper scaling can be used as initial estimates for block matching. The auxiliary stream frames can be estimated in two ways by using the following coherence equation [31]:

$$\boldsymbol{v}_m + \delta_t = \boldsymbol{v}_a + \delta_{t+k} \qquad \text{(EQ 4.3)}$$

where $\boldsymbol{v}_m$ is the main stream motion vector of a segment between the frames at instants t and $(t+k)$, $\boldsymbol{v}_a$ is the auxiliary stream motion vector between the frames at instants t and $(t+k)$, δ_t is the left-right disparity at instant t, and δ_{t+k} is the left-right disparity at instant $(t+k)$. The auxiliary frame corresponding to the segmented frame can be estimated by reversing the disparity map obtained during MDBS. The other auxiliary stream frames are estimated through DCP. For every segment in the $(t+k)$th main stream frame, a best match in the corresponding auxiliary frame is found. Then the direction of prediction is reversed to estimate the auxiliary frame. For small k, δ_t can be used as a good initial estimate for δ_{t+k}. Since the disparity map for each frame is available at the decoder, this case comes under configuration-1 and we would refer to the scheme as ST-1 (segment tracking - configuration 1). The frame structure is illustrated in Fig. 4.7. A similar extension can also be realized by using motion compensation to predict the B_A-frames.

Since all B-frames and B_A-frames are estimated by reversing the prediction direction, these frames will have regions with no predictions and multiple candidate predictions at overlaps. The multiple matches can again be resolved based on the disparity estimates. However, the filling-in procedure is not as simple as in RDBS. This is because the frames are now offset in time, and hence a simple 1-D prediction along scan lines is not possible. Also, since the main sequence has to be coded at a higher quality, simple concealment is not sufficient. The increase in residual coding overhead at high bit-rates can more than offset the advantage gained by distributing the segmentation overhead over a group of frames. However, computationally this scheme is quite attractive. This is because the segmentation frequency is significantly reduced and the motion and disparity compensation complexities are also considerably reduced by using the suitably scaled past estimates as initial estimates. The latter is possible only because the same segment is *tracked* over time and across views. Also, the B-frames and B_A-frames need not be decomposed multiresolutionally, as refinements from the initial estimates can be carried out at the finest resolution level itself. Hence this scheme will be ideal in situations where a very high quality main



MDBS - *motion and disparity based segmentation*

Track - The segments in the reference frame are *tracked* and the motion/disparity vectors are *reversed* to predict the frame. The *holes* are then filled in.

MCP^1 - MCP w.r.t to reference frames (for under-compensated blocks).

MCP^2 - MCP of the segments

FIG. 4.7: Segment tracking scheme ST-1 - configuration 1

stream is not required, or in cases where a lower computational complexity is desired.

A half-pixel accurate prediction in the reference frame is obtained for each tracked segment. As in RDBS, we need two bits per direction to represent this half-pixel accurate estimate. The holes in this case are filled-in by first extracting their locations and then performing MCP. This is done to exploit the fact that the holes typically are quite long in one direction, so that only a few motion vectors are needed. Also, if a particular order is employed in extracting the holes, the decoder can repeat that order without any ambiguities, and hence no location coding overhead is incurred. The residuals in the auxiliary frames are coded using the quadtree-based VQ/SQ combination, with bidirectional motion estimation at the 8x8 block size to exploit the temporal redundancies that were not exploited during segment tracking.

4.6 MIXED-RESOLUTION BASED CODING

Low bit-rate coding is desired for the auxiliary sequence to reduce the excess bandwidth. This restricts the number of bits that can be allocated for residual coding. The significant residuals that are left uncoded can result in visually distracting artifacts. The noticeable artifacts can be suppressed by trading off resolution and coding the auxiliary stream frames at a reduced resolution. Psychophysical studies [12, 30, 89, 107] have shown that satisfactory stereoscopic perception is achieved even when one of the stereoscopic sequences is presented to a viewer at a

reduced resolution. Based on psychophysical experiments with random-dot stereograms, Julesz [12] has reported that stereopsis can occur even if spatial similarities exist only in a particular frequency band. Based on an experiment where a sharp image was presented to the right eye and a significantly blurred image was present to the other eye, he reports [12, page 96] that "the stereoscopic image pair is easy to fuse, and the binocular percept appears not only in depth but seems as detailed as the sharper image".

Mixed-resolution based stereoscopic image coding was first described by Perkins in [30], where each 4x4 block in one view is averaged to obtain one pixel at the reduced resolution. During display, a bilinear interpolation is applied to expand the size. The subsampling and upsampling are thus done in an adhoc fashion, without any consideration about aliasing or the reconstruction quality. A Gaussian pyramid (see Section 2.3.1) based subsampling and upsampling is used for reduced resolution coding in [89]. Since we employ a multiresolution framework for segmentation and motion/disparity estimation, the mixed-resolution based coding automatically fits into our framework. The multiresolution estimation of motion or disparity needs to be carried out only up to the resolution desired.

Figure 4.8 shows the modifications needed at the encoder and decoder for mixed-resolution coding, with the auxiliary sequence being coded at half the actual horizontal resolution. Since the residual coding overhead is smaller at a reduced resolution than at the original resolution, the bits

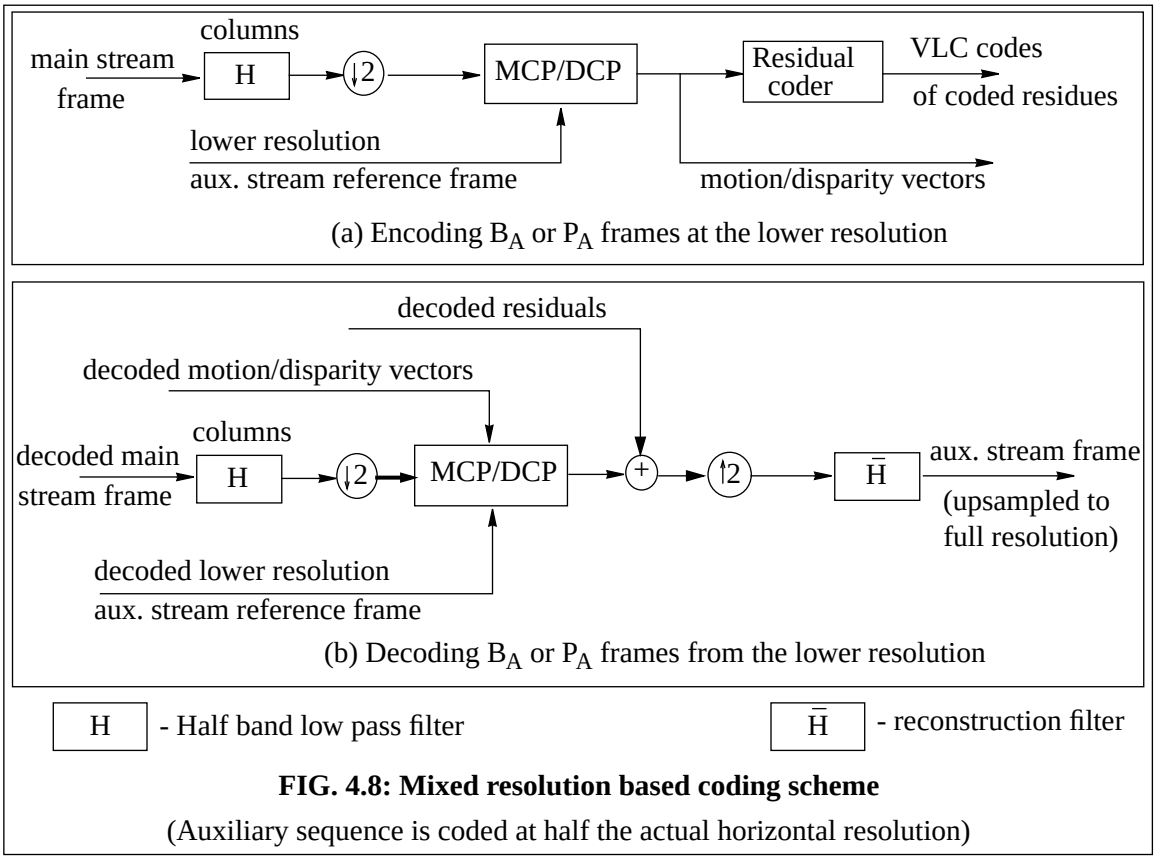


FIG. 4.8: Mixed resolution based coding scheme

(Auxiliary sequence is coded at half the actual horizontal resolution)

available for residual coding can be used for suppressing significant artifacts. Also, as the intra-coded frame typically discards most high frequency components, the loss of information as compared to full resolution coding can be expected to be small. However, the reduction in the horizontal resolution can result in a reduction in depth plane resolution or “stereo-acuity” [35]. To avoid this, we employ a sub-pixel-accurate disparity estimation at the reduced resolution that is equivalent to a half-pixel accurate disparity estimation at the original resolution. Since the filters are non-ideal and the high frequency components are lost, the reconstruction can contain aliased high frequency components (if the original image had significant energy at the high frequencies). A Wiener filter for reducing such aliasing-noise has been used in [15]. The decoder complexity increases because of the need for filtering, upsampling and downsampling. However, as we mentioned in Section 2.3.6, to achieve spatial and temporal scalability multi-rate filter banks are in general desirable in decoders. The hardware resource available for this purpose can be used for mixed-resolution coding. Thus, mixed-resolution coding provides a method for trading resolution for perceived quality in a controlled fashion, which could be a significant factor in making stereoscopic video transmission practical.

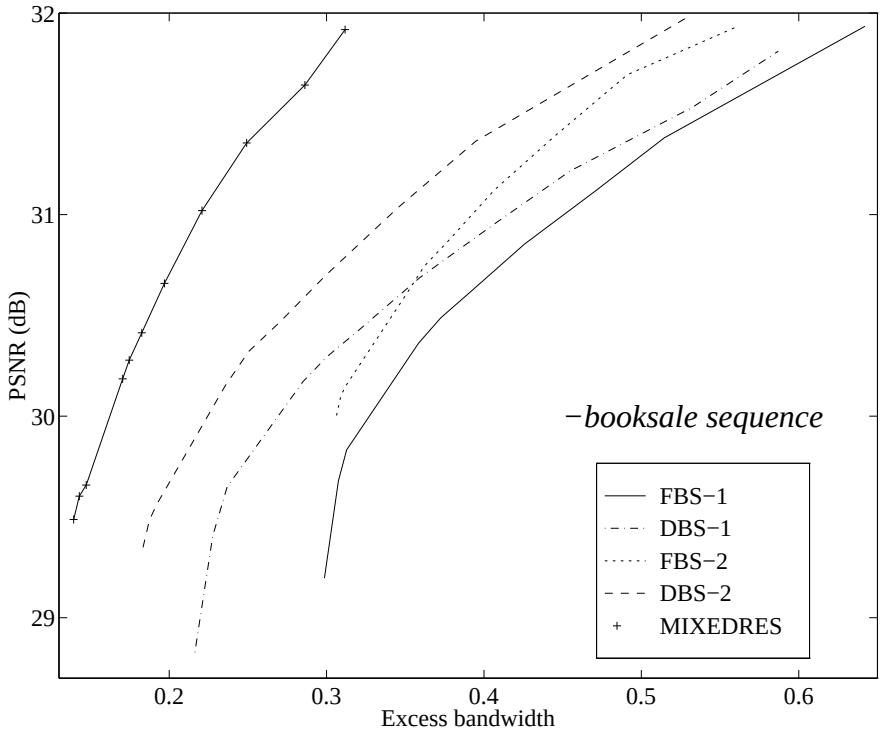
4.7 RESULTS

The two baseline schemes and the five extensions were tested over the *booksale*, *crowd*, *aqua*, *train*, *tunnel* and *piano* sequences described in Appendix A. The fields in the sequences are considered as frames during motion compensation[1]. For the field sequential sequences (*booksale* and *crowd*) where the successive fields for the same eye are 1/30th of a second apart, the distance between reference frames is chosen as M=4 fields, and the I-to-I frame distance is chosen as N=16 fields. For the DISTIMA sequences where the successive fields for the same eye are 1/50th of a second apart, M=7 fields and N=28 fields are chosen.

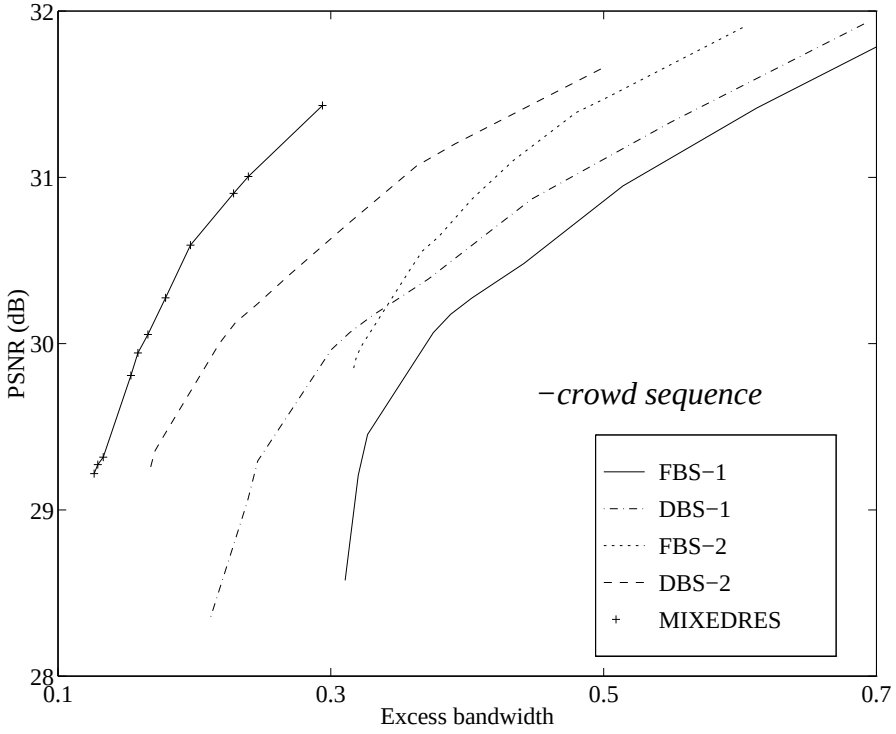
The main stream frames are coded at a high PSNR in all the cases discussed. The auxiliary stream’s quality, and correspondingly its bit-rate, are controlled using the N_{max}, T, and E_{max} parameters of the residual coder. For the segmentation schemes, the segmentation parameters are fixed (at an optimal point near the knee of the R-D curves such as the ones shown in Figure 3.8) across the different trials. The bit-rate (R) and distortion (D) measures used to compare the different schemes are the *average bpp* and the *average PSNR* (defined in Figure 2.2.8) respectively.

The rate-distortion (R-D) curves for the FBS-1, DBS-1, FBS-2, DBS-2 and mixed-

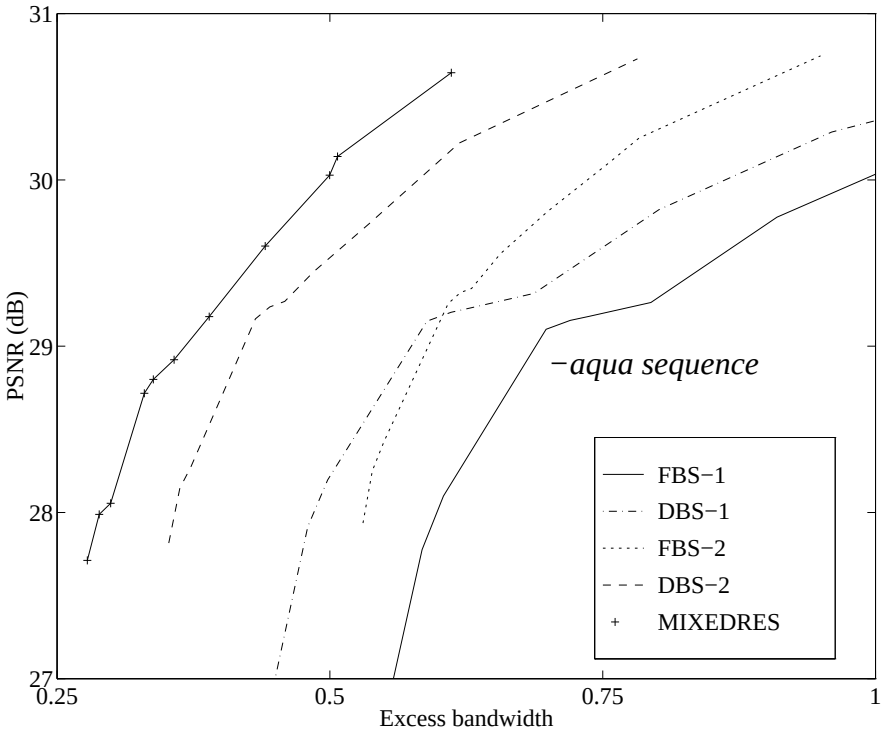
1. The MPEG-2 standard uses field prediction and dual-prime [25] prediction modes in addition to frame prediction, in order to exploit the redundancy between adjacent fields in an interlaced sequence. For simplicity, we employ only frame prediction and treat each field as a frame. Hence field and frame will be used interchangeably in this thesis.


32
31
30
29
PSNR (dB)
0.2
0.3
0.4
0.5
0.6
Excess bandwidth
−booksale sequence
FBS−1
DBS−1
FBS−2
DBS−2
MIXEDRES

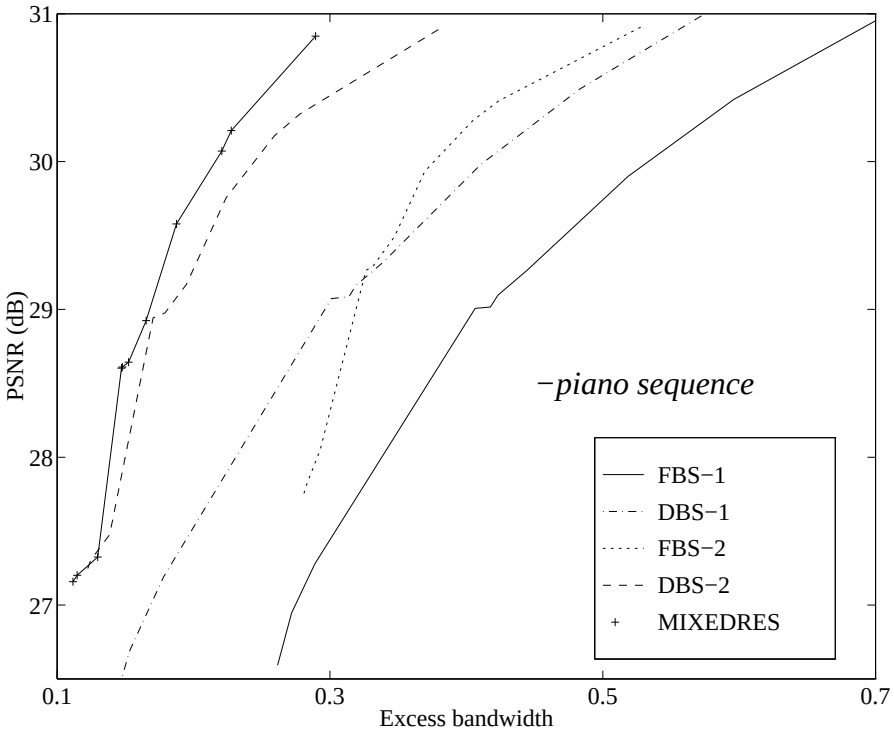
(a)


32
31
30
29
28
PSNR (dB)
0.1
0.3
0.5
0.7
Excess bandwidth
−crowd sequence
FBS−1
DBS−1
FBS−2
DBS−2
MIXEDRES

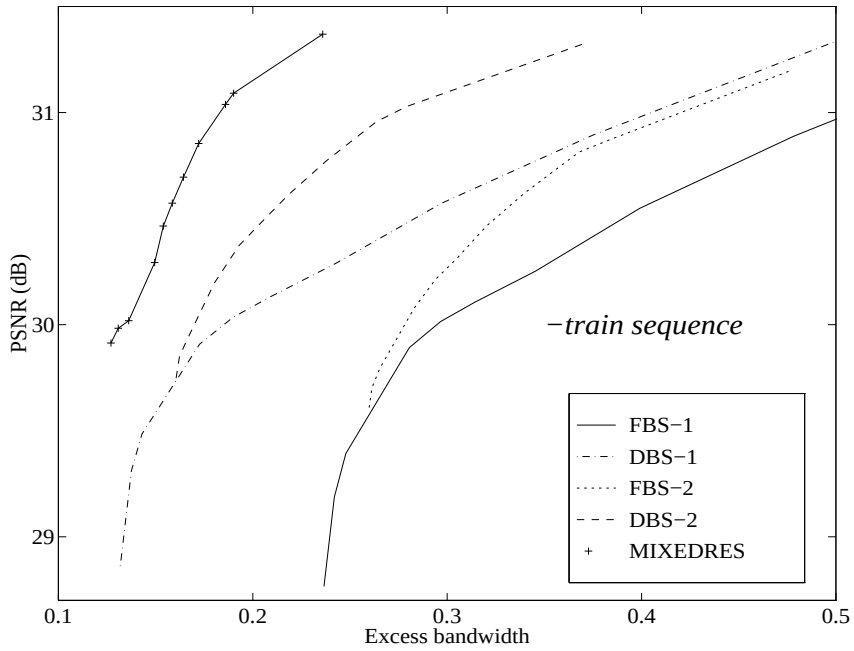
(b)


31
30
29
28
27
PSNR (dB)
0.25
0.5
0.75
1
Excess bandwidth
−aqua sequence
FBS−1
DBS−1
FBS−2
DBS−2
MIXEDRES

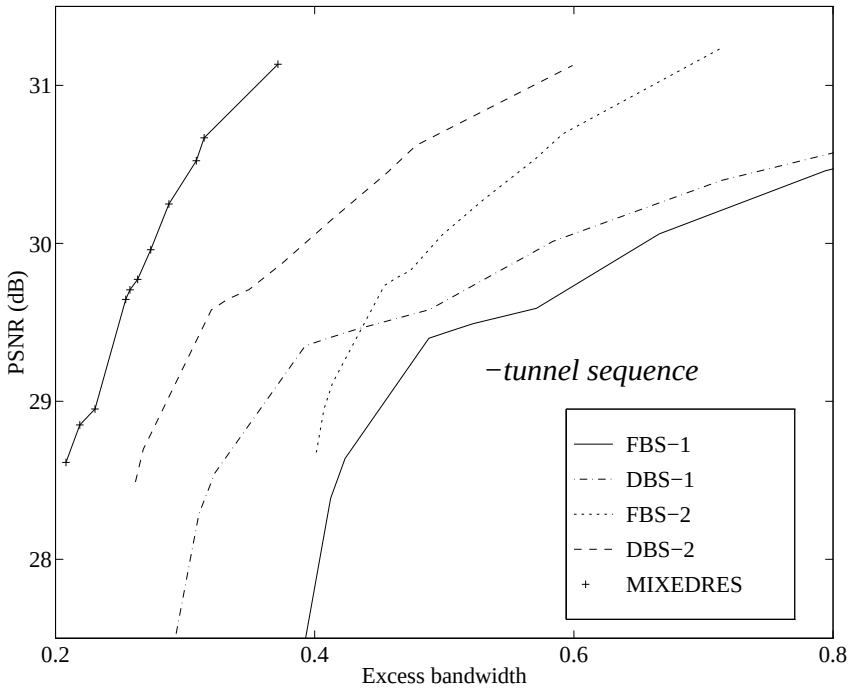
(c)


31
30
29
28
27
PSNR (dB)
0.1
0.3
0.5
0.7
Excess bandwidth
−piano sequence
FBS−1
DBS−1
FBS−2
DBS−2
MIXEDRES

(d)



(e)



(f)

FIG. 4.9: (a)-(f) Rate-distortion performances of the FBS-1, DBS-1, FBS-2, DBS-2 and mixed-resolution (using DBS-2) schemes for the auxiliary sequences of six stereoscopic test sequences

Excess bandwidth and the PSNR are varied by varying the quality thresholds in the residual coder. The main sequence quality is maintained the same during the different trials (see Table 4.2).

TABLE 4.2: Main sequence bit-rate and quality for the different extensions

Test Sequences	Field or Frame size (pixels)	Frame rate per eye (fps)	FBS-1/DBS-1/FBS-2/ DBS-2/Mixed resolution			RDBS		ST-1	
			bit-rate		PSNR (dB)				
			bpp	Mbps		bpp	PSNR	bpp	PSNR
Booksale	640 x 240	30	0.477	2.198	34.48	0.462 -0.015	34.36 -0.12	0.487 +0.001	34.14 -0.34
Crowd	640 x 240	30	0.471	2.170	33.98	0.469 -0.002	33.81 -0.17	0.475 +0.04	33.61 -0.37
Aqua	720 x 576	25	0.312	3.235	33.33	0.271 -0.041	33.25 -0.08	0.406 +0.094	32.93 -0.40
Piano	720 x 576	25	0.475	4.925	34.77	0.430 -0.045	34.61 -0.16	0.459 -0.016	34.2 -0.57
Train	720 x 576	25	0.554	5.744	33.72	0.500 -0.054	33.73 +0.01	0.561 +0.007	33.35 -0.37
Tunnel	720 x 576	25	0.378	3.919	33.34	0.354 -0.024	33.23 -0.11	0.424 +0.046	32.96 -0.38

resolution schemes are shown in Fig. 4.9(a)-(f). The auxiliary stream frames for the mixed-resolution case are coded at half the original horizontal resolution using the DBS-2 scheme. In all these five dependent-coding cases, the main sequence frames are coded at the same PSNR (shown in Table 4.2). To enable easy comparison in terms of excess bandwidth, the rate is shown in terms of excess bandwidth on the R-D plots. Some of the observations from these plots are:

[1] DBS-1 outperforms FBS-1 at all the quality settings and for all the sequences. The decrease in the bit-rate for the DBS-1 scheme compared to the FBS-1 scheme, for a given quality, is primarily due to the reduction arising from the segmentation. As such, the savings in bpp remains more or less constant across the different quality levels. In other words, the percentage savings in bit-rate is higher at low bit-rates.

[2] DBS-2 outperforms FBS-2 at all quality settings and for all the test sequences. Again, the bit-rate reduction at a given quality is only due to the gains from segmentation, as all other factors are maintained the same. In addition, DBS-2 also consistently outperforms DBS-1 and FBS-1. This can be attributed to the bidirectional prediction of P_A and B_A-frames. This can also be seen from the fact that FBS-2 performs better than FBS-1 in all the cases. Also, FBS-2 performs better than DBS-1 at higher bit-rates. Thus, configuration-2 performs better in the rate-distortion sense. But as the disparity map is not available at the decoder for this

configuration, intermediate views cannot be synthesized.

[3] The bit-rate using FBS methods cannot be reduced below a certain minimum bit-rate (which depends on the average number of motion or disparity coding bits per blocks). This implies that the differential predictive coding of motion and disparity is not useful beyond a certain point, due to the presence of spurious matches. On the other hand, by varying the segmentation parameters, arbitrarily low bit-rates for the segmentation schemes can be achieved.

[4] The PSNR shown for the mixed-resolution based coding case is at the lower resolution. This curve has been shown to emphasize the fact that by trading off resolution, we can improve the quality of residual coding compared to what is achieved at the original resolution. The bit-rate savings is higher at higher quality settings indicating that more bits are being spent in the DBS-2 case to encode the residuals at the edges, which are not present at the lower resolution. Psychophysically, suppression of artifacts might outweigh loss of resolution.

[5] The excess bandwidth is anomalously high for the *aqua* sequence in all cases, as its main sequence is coded at a low bit-rate owing to the very slight motion between frames. This supports our earlier statement that the excess bandwidth cannot be fixed as a percentage of the main sequence bandwidth.

TABLE 4.3: Excess bandwidth (%) comparison at a fixed PSNR (good quality)

Sequences	PSNR[a] (dB)	Excess bandwidth[b] (% of main stream bandwidth)				
		FBS-1	DBS-1	FBS-2	DBS-2	Mixed Resolution
Booksale	31.5 (-2.98)	52	49 (-3)	44 (-8)	39 (-13)	25 (-27)
Crowd	31.0 (-2.98)	52	47 (-5)	42 (-10)	35 (-17)	24 (-28)
Aqua	29.8 (-3.53)	91	79 (-12)	69 (-22)	55 (-36)	46 (-45)
Piano	30.4 (-4.37)	59	47 (-12)	42 (-17)	29 (-30)	25 (-34)
Train	30.9 (-2.82)	48	38 (-10)	40 (-8)	26 (-22)	18 (-30)
Tunnel	30.5 (-2.84)	83	79 (-4)	57 (-26)	47 (-36)	31 (-52)

a. The difference between the main sequence PSNR and the auxiliary sequence PSNR is shown within parenthesis. (compare with Table 4.2)

b. The reduction in excess bandwidth w.r.t the scheme with the highest excess bandwidth at the given PSNR is shown within parenthesis.

TABLE 4.4: Quality comparison at a low excess bandwidth

Sequences	Excess bandwidth (%)	PSNR[a] (dB)				
		FBS-1	DBS-1	FBS-2	DBS-2	Mixed Resolution
Booksale	30	29.2	30.3 (+1.1)	30.0 (+0.8)	30.7 (+1.5)	31.8 (+2.6)
Crowd	32	29.1	30.1 (+1.0)	30.0 (+0.9)	30.8 (+1.7)	31.5 (+2.4)
Aqua	46	26.7	28.9 (+2.2)	28.4 (+1.7)	29.8 (+3.1)	30.3 (+3.6)
Piano	28	27.1	28.9 (+1.8)	27.8 (+0.7)	30.3 (+3.2)	30.8 (+3.7)
Train	26	29.7 (+0.1)	30.4 (+0.8)	29.6	30.9 (+1.3)	31.4 (+1.8)
Tunnel	40	27.6	29.3 (+1.7)	28.7 (+1.1)	30.1 (+2.5)	31.2 (+3.6)

a. The increase in PSNR w.r.t the scheme with the lowest PSNR at the given excess bandwidth is shown within parenthesis.

The excess bandwidths needed to code the auxiliary sequences at a particular high PSNR value are compared for the five dependent-coding methods in Table 4.3. However, as we are more interested in the low excess bandwidths, we compare the PSNRs at a fixed low excess bandwidth in Table 4.4. While the percentage reduction in excess bandwidth by using segmentation methods is not large at high quality settings, the improvement in PSNR at a fixed low excess bandwidth ranges from 1 to 2 dBs over the FBS counterparts.

In Fig.s 4.10 and 4.11, we show the percentage split between the number of blocks that are compensated using disparity and motion estimation, for the FBS-2 and DBS-2 schemes. It can be seen that the percentage of blocks that use DCP is a little lower than the percentage of blocks using MCP. Also, as the quality of the auxiliary frames increase, the DCP percentage drops further. This justifies our discussion in Section 4.1.2 about the effect of the quality of the reference frames on coding. The general reduction in the percentage of blocks using DCP can be because, (1) a finite disparity exists for all points that are not very distant from the cameras, as opposed to motion, which is present only for regions undergoing displacement from frame-to-frame, and (2) the extent of disparity between views is typically greater than the extent of frame-to-frame motion. In our experiments, we used only distortion measures to decide between DCP and MCP. However, as disparity is a scalar for a parallel-axes imaging geometry, a rate-and-distortion based criterion may favor DCP more.

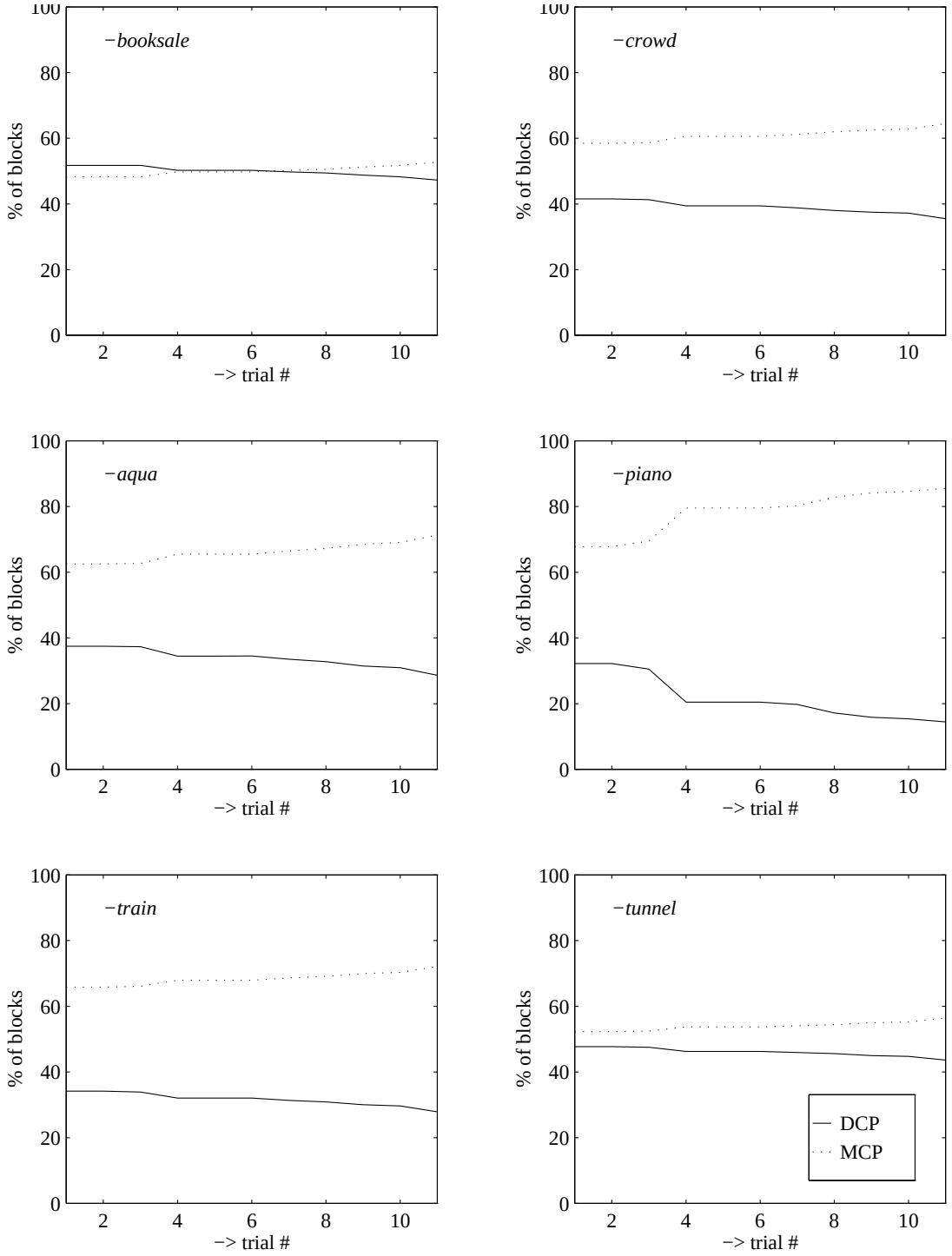


FIG. 4.10: Percentage split between (8x8 sized) P_A- and B_A-frame blocks predicted using disparity and motion compensation for the FBS-2 scheme

Increasing trial numbers correspond to increase in the quality of the auxiliary sequence frames. It can be seen that MCP is favored more as the quality of the auxiliary sequence increases. Except for the *piano* sequence where the two cameras are not matched, DCP is used for about 40% of the blocks.

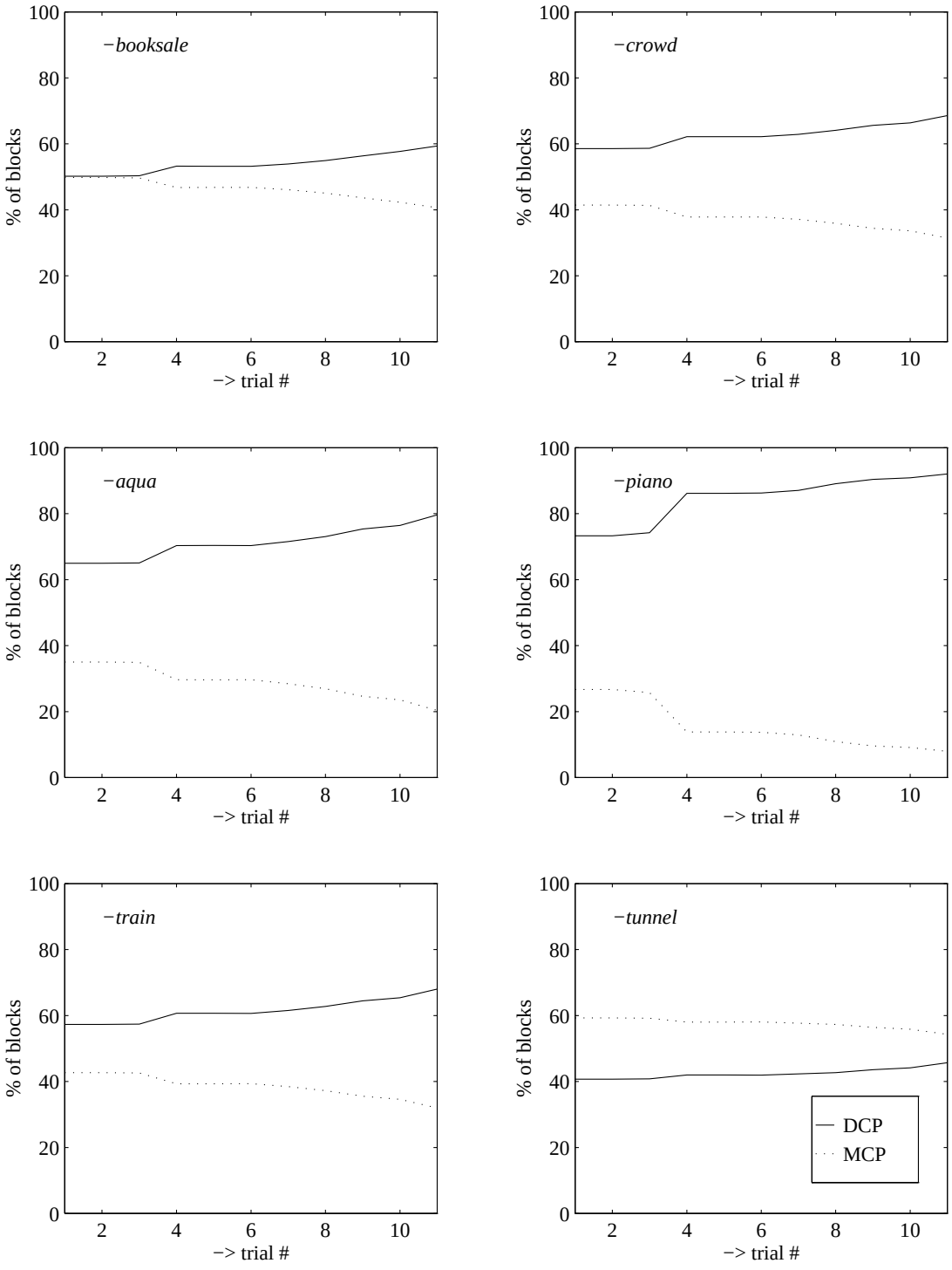


FIG. 4.11: Percentage split between variable-sized blocks in the B_A- and P_A-frames that are predicted using disparity and motion compensation

(Same observations as in Fig. 4.10 hold good)

For the *piano* sequence, the percentage of blocks using DCP is anomalously low due to the fact that the left and right cameras for this sequence are not well matched, thus favoring inter-view coding more than intra-view coding.

In Fig. 4.12 we compare the performance of the RDBS joint-coding scheme against the performances of other configuration-1 schemes, namely, FBS-1 and DBS-1. The observations are:

[1] RDBS performs better than FBS-1 for all cases.

[2] Except for the *booksale* and *crowd* sequences, RDBS performs as well as or better than DBS-1 for the auxiliary sequence. The improvement in performance is because the segmentation coding overhead is shared by the main and auxiliary sequences. Because of the field-sequential nature of *booksale* and *crowd* sequences, the disparity computed using adjacent left and right fields can have a 2D motion component as well. Erroneous predictions for exposed regions using the scan-line based prediction method described in Section 4.5.2, can be a reason for the reduction in R-D performance, as this method assumes that the disparity has only a horizontal component.

[3] Since the main sequence in RDBS is coded based on segmentation, a slight improvement in the rate-distortion performance for the main sequence over FBS-based coding can be seen. The improvements are tabulated in Table 4.2.

Figure 4.13 compares the performance of ST-1 with those of DBS-1 and FBS-1. The main stream R-D values are given alongside the plots and are tabulated for comparison in Table 4.2. The observations are:

[1] The R-D performances of ST-1 and DBS-1 for the main sequence are more or less similar, except for the *aqua* sequence. The *aqua* sequence has a very high spatial detail. The increase in bit-rate may be due to the need to maintain a high quality for the main sequence.

[2] For the auxiliary stream, the R-D performance is very close to that of DBS-1 in most cases. Since the segmentation overhead is shared by several frames in the ST-1 case, the performance at low bit-rates is very close to or better than the performance of the DBS-1 scheme. However at higher PSNRs, the distortion introduced by the filling-in procedure results in a poorer performance than DBS-1 for the *crowd*, *piano*, and *tunnel* sequences.

However, the ST-1 scheme results in significant computational simplifications due to reduced frequency of segmentation and simplified motion and disparity estimation when compared to the DBS-1 scheme. The disparity estimation using the coherence equation (see Section 4.5.3), requires only a search at the original resolution level over a very small search range (typically ± 2 pixels horizontally). Hence, ST-1 would be useful for applications where a low computational complexity is preferred.

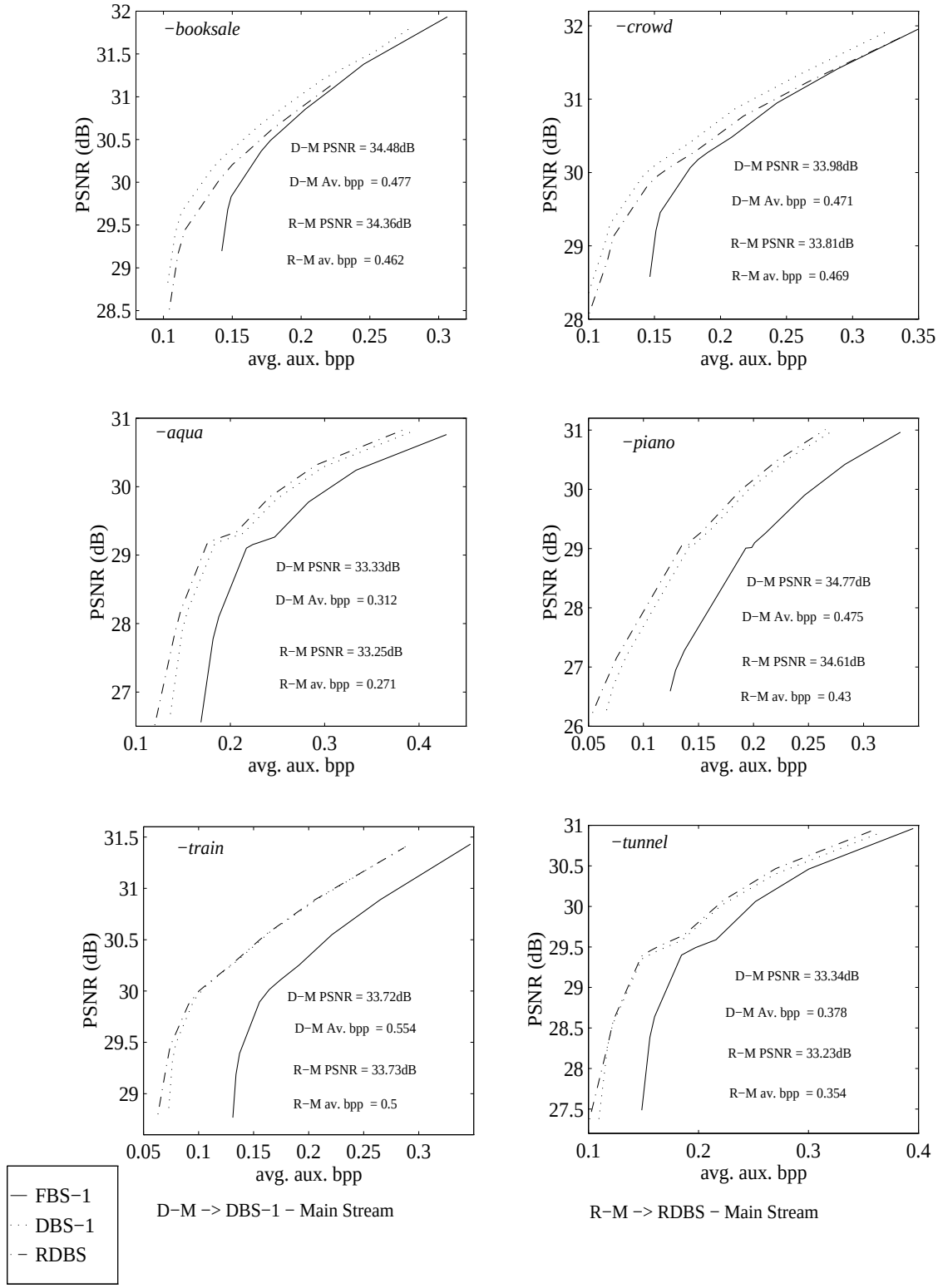


FIG. 4.12: Rate-distortion performance of the RDBS joint-coding scheme for the auxiliary sequences of the six stereoscopic test sequences

Compared here with the R-D performances of FBS-1 and DBS-1 schemes. The main sequence quality and bit-rate were maintained across the trials and are shown in Table 4.2.

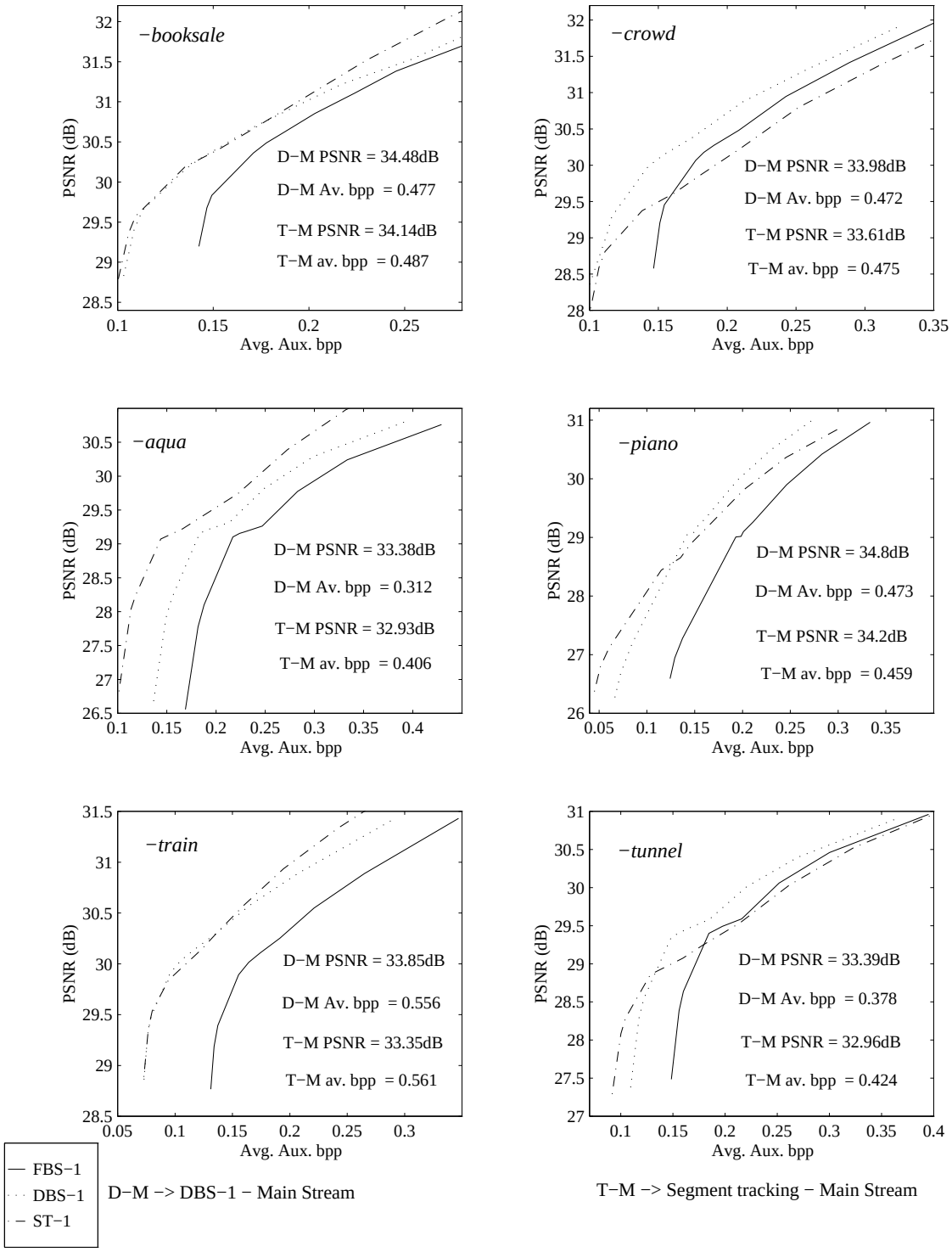


FIG. 4.13: Rate-distortion performance of the ST-1 joint-coding scheme for the auxiliary sequences of the six stereoscopic test sequences

The performance is compared with the R-D performances of the FBS-1 and DBS-1 schemes. The main sequence PSNR and bit-rate for the DBS-1 and ST-1 schemes are also shown in the plots for comparison.

4.8 SUMMARY OF SEQUENCE CODING EXTENSIONS

In this chapter, we have introduced a framework for coding stereoscopic sequences that reduces the excess bandwidth needed to code stereoscopic video at a reasonable perceived stereoscopic quality, while maintaining quality compatibility with existing monoscopic transmission, by taking advantage of cross-stream correlations and by using content-adaptive coding, reduced quality coding and the psychophysically-motivated mixed-resolution coding methods. Two different configurations of coders were described depending on whether a complete disparity map for each stereo-frame is available at the decoder. In addition to simple FBS-based baseline schemes and their straightforward segmentation-based extensions, we have also developed two joint coding schemes to improve primarily the computational performance, and if possible, the coding performance. We summarize these coding extensions in Table 4.5 and highlight their features in Table 4.6 for easy reference.We have also developed a scheme for predicting perspective-based occlusions using the imaging geometry and the estimated disparity map.

TABLE 4.5: Summary of stereo sequence coding extensions

	Extension	Main stream (B-frame)	Auxiliary Stream (B-frame)
Configuration 1	FBS-1	FBS-based MCP (8x8 blocks)	FBS-based MCP (8x8 blocks)
	DBS-1	FBS-based MCP (8x8 blocks)	DBS based DCP
	RDBS	MCP of DBS blocks	Predicted by reversal of disparity map
	ST-1	Motion based tracking of the reference frame MDBS blocks	Disparity based tracking of DBS blocks in the reference frame. Simplified search using the coherence equation.
Configuration 2	FBS-2	Same as FBS-1	Bidirectional prediction: best of MCP/DCP for each 8x8 block
	DBS-2	Same as FBS-1	Bidirectional prediction: best of MCP/DCP for each DBS block
Other	Mixed Resolution	Any of the above.	Any of the above at a reduced resolution. Residuals are coded at that reduced resolution.

TABLE 4.6: Contrasting features of the stereo sequence coding extensions

Extension	Features
FBS-1	Datum for configuration-1 - Simple MPEG-like implementation - Non-adaptive - due to spurious matches the disparity map is not smooth - higher disparity coding overhead
DBS-1	Adaptive block size - smooth disparity map - sum of disparity coding bits and segmentation overhead is less than disparity coding bits in FBS-1 - one segmentation per stereo frame
RDBS	Segmentation based coding of both streams - DBS blocks further segmented based on motion compensation - one segmentation per stereo frame - smooth disparity map - main stream bit-rate typically lower than for FBS-1/DBS-1 - auxiliary stream requires filling in of holes due to reversal of prediction direction
ST-1	Segmentation based coding of both streams - joint motion and disparity based segmentation - one segmentation per reference stereo frame - both streams require filling in of holes - reduced computational complexity due to the use of previous estimates of motion and disparity to reduce search area - decoder uses the same set of segments to reconstruct all B-frames - typically suffers a penalty in quality due to reversal of prediction direction in both streams.
FBS-2	Datum for configuration-2 - Simple MPEG-like implementation - usually better than FBS-1 as bidirectional prediction is used - decoder does not have a complete disparity map
DBS-2	Adaptive block size (based on either motion or disparity) - usually better than FBS-1, DBS-1 and FBS-2 in terms of auxiliary stream compression.
Mixed Resolution	Sacrifices resolution to achieve lower bit-rates - psychophysically motivated - fits well within the multiresolution framework

We have shown through experiments over six stereoscopic test sequences, with different scene contents, types of motion, and extent of disparities, that the segmentation-based extensions perform better than FBS-based methods. At low excess bandwidths, the operating region in which we are more interested, the rate or distortion improvement achieved using our segmentation methods is significant. We have also extended the framework to include mixed-resolution based coding, which can provide significant reduction in excess bandwidth without causing a significant

difference in perceived stereoscopic quality (as demonstrated through psychophysical experiments in [30, 89]).

Our reported results are based only on the PSNR quality metric. Though the subjective quality was assessed by a few members of our research group and many visitors, formal psychophysical experiments will be needed to substantiate the relationship between our demonstrate PSNR quality improvement and the apparent perceived quality improvement with our approach.

Chapter 5

Multi-view compression and Synthesis of intermediate views

5.1 INTRODUCTION

In Chapters 3 and 4 we restricted our discussion to compression of 2-view stereoscopic images and sequences, obtained from a pair of cameras that are offset horizontally by the nominal human interocular separation. These 2-view images and sequences provide the relative depth information only from one pair of viewpoints; thus they are suited for only one viewer and one viewing location[1]. To provide the correct perspective to each viewer in a multi-viewer scenario, multiple views need to be captured and transmitted. In addition to binocular parallax, depth can also be discerned from motion parallax associated with head movements. The lack of motion parallax during head movements can cause confusion and a perception of unnaturalism when stereoscopically viewing a fixed pair of perspectives. By showing the correct perspectives depending on the position of the head (or eyes) motion parallax cues or the sense of 'look-around' can be provided to a viewer. Hence, multiple views are required in this situation also.

The nature of the views needed in these two applications is different. For the multi-viewer case we need multiple pairs of views, with a nominal interocular separation within a view-pair and a large separation (corresponding to the separation between adjacent viewers) between view-pairs. Because of this, the correlation between view-pairs is small compared to the correlation within a view-pair. On the other hand, for providing motion parallax during head movements, a set of views centered around the mean position of the viewer is required. Though a continuum of views are needed in this case, the typical solution is to acquire a discrete set of views from which the intermediate views can be synthesized with reasonable accuracy. Typically a linear array of cameras on a horizontal line with constant separation between them is employed for this purpose (Fig. 5.1). In this thesis, we do not consider multi-view compression for the multi-viewer scenario; we restrict our attention to the compression of multiple views from the equally spaced camera configuration and the associated synthesis of intermediate views.

The ability to synthesize intermediate views is also useful in providing variable baselines

1. Stereopsis can be achieved from a continuum of viewing locations; but the perspectives are incorrect except when viewed from the particular location for which the views were created.

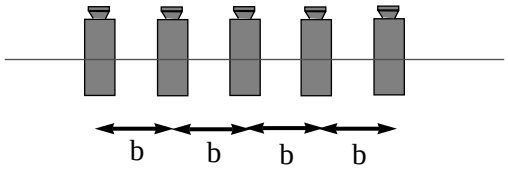


FIG. 5.1: Equi-spaced multi-camera configuration for providing 'look-around'

between the left and right eye views. Since the interocular separation is different for different people, such a setup would help the viewers to change the baseline to suit their comfort[1]. The range over which views need to be synthesized in this case is typically smaller than the range required to provide motion parallax during head movement.

In this chapter we provide a brief survey of earlier work in Section 5.2 and suggest possible means of extending the 2-view sequence methods of Chapter 4 to multiple views in the subsequent subsections. We present a simple procedure for synthesizing intermediate views in Section 5.3. Using this procedure we evaluate the suitability of the disparity maps estimated using our disparity-based segmentation method and the fixed block-size based disparity compensation method. In Section 5.4 we consider multi-view extensions of the two-view sequence compression methods presented in Chapter 4. In particular, we present a 3-view coding scheme for which the performances can be extrapolated from the performances of the 2-view coding methods; the 3-view configuration has certain potentially attractive features. We also consider an adaptation of the multiple-baseline disparity estimation method, which can reduce ambiguous matches, for coding multi-view images.

5.2 PRIOR WORK

Multi-view coding and synthesis of intermediate views has been studied by several researchers. A pel-recursive disparity compensation based coding of the multiple views was presented in [86]. In that paper, one view is coded separately and all the other views are predicted from the independently coded view. An analysis of epipolar plane images (EPIs), which are a set of multiple views very closely spaced, is presented in [85]. The analysis is used for obtaining a compact representation of the unambiguous depth information over a viewing range. However, the acquisition, processing and compression of these very large number of views may not be practical for transmission purposes. Recently, a variation of the EPI analysis method was presented for compression purposes in [93]. The multi-view images are stacked up in what they call the *multi-view image space*. The disparity between adjacent views is represented in a *normalized object space*. A triangular mesh for the scene is chosen based on intensity variance. Using affine

1. However, as only one baseline can be specified per display, this feature would be useful only in a single viewer situation (unless all viewers agree on the same baseline).

transformation based matching, the depth for each grid node on the mesh is calculated in the normalized object space. Based on this disparity representation and the texture information of each triangular patch, intermediate views are synthesized. Though the method yields high compression ratios for simple scenes, it does not scale well with complex scenes. A triangular 3-camera setup has been used in [32] to minimize occlusions typically incurred with a horizontally separated 2-camera setup. In that paper, the disparity is estimated using dynamic programming techniques for intensity edges; this sparse disparity is interpolated to obtain a dense disparity map which is used to synthesize intermediate views. Good synthesis capability has been reported for simple scenes. However, the paper does not discuss the coding approach used to code the disparities. A coding scheme is reported in [95], where two extreme view images are coded along with two disparity maps that specify the disparity for each pixel in these extreme views. In addition, if regions occluded in the extreme views are visible in the intermediate views, an additional intensity image and a disparity map are coded. The disparities are estimated using a dynamic programming approach constrained by the parallel axes camera geometry. As in [32], this paper does not discuss the coding of the dense disparity maps. An approach for adaptively selecting optimal reference frames for predicting multi-view images (in the sense of maximizing the correlation or minimizing the occlusion) along with a robust intermediate view synthesis method that refines the disparity maps obtained from block disparities at the decoder is presented in [122].

5.3 INTERMEDIATE VIEW SYNTHESIS (IVS)

In this section we present an algorithm for synthesizing intermediate views, given two views and a unidirectional disparity map between the two views. Let L and R be the two views given. We assume that a disparity map D obtained by performing disparity compensated prediction of L from R is available. Let b be the baseline distance between the optical centers of the two cameras used to acquire L and R. With pin-hole approximations for the cameras and a parallel axes binocular imaging geometry, the disparity δ_{lr} between corresponding image points P_L and P_R on corresponding scan-lines in the left and right views is given by,

$$\delta_{lr} = \frac{bf}{z} \qquad \text{(EQ 5.1)}$$

where f is the distance between the imaging plane and the pin-hole, and z is the distance between the lens plane and the real world point P. Now if we consider an intermediate virtual camera between the left and right cameras at a distance of x from the left camera, then the disparity δ_{li} between P_L and the corresponding point in the intermediate view P_I is,

$$\delta_{li} = \frac{xf}{z} = \left(\frac{x}{b}\right)\delta_{lr} = \alpha\delta_{lr}; \quad 0 \le \alpha \le 1 \qquad \text{(EQ 5.2)}$$

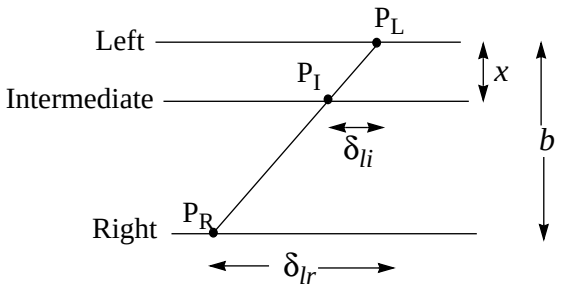


FIG. 5.2: Equivalent representation when the images are aligned width-wise

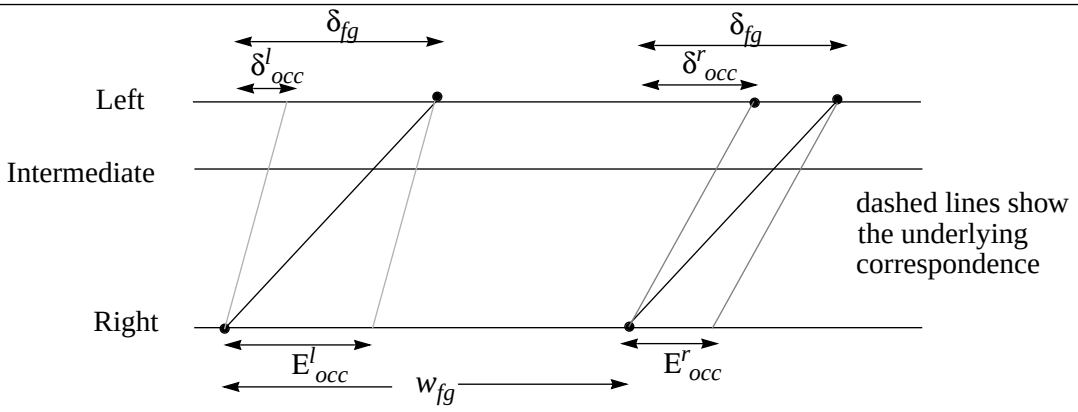
Hence, if x_l is the position of P_L and x_r is the position of P_R in the corresponding and left and right view scan-lines, the position of P_I in the intermediate view scan-line can be obtained as,

$$x_i = (1-\alpha)x_l + \alpha x_r = x_l - \alpha\delta_{lr} = x_r + (1-\alpha)\delta_{lr} \quad \text{(EQ 5.3)}$$

This implies that P_I lies on the line joining P_L and P_R when the left and right scan-lines are aligned width-wise separated by the baseline distance as illustrated in Fig. 5.2; x_i can be predicted given x_l and δ_{lr}. Thus, if a point is unoccluded in both the left and the right views, then the intermediate view point can be obtained in a straightforward fashion based on the knowledge of the left-right disparity[1]. However, due to occlusions, disparity is not defined at all points. This is further complicated by the fact that when estimating one view from the other, the pixels occluded in the reference frame but unoccluded in the view being estimated are assigned erroneous disparity estimates. For example, in Fig. 5.3, if the left view is being estimated from the right view, the region to the left of the occluding object may be assigned a disparity estimate that is larger than the disparity of the foreground object. In addition, the constant-disparity assumption based block matching and the aperture problem associated with block size can result in further errors in the estimated disparity map.

As can be seen from Fig. 5.3, the intermediate view pixels corresponding to occluded regions can be predicted in most cases from one of the views if their underlying disparity estimates are known (except when certain regions are occluded in both the left and right views but should be visible in an intermediate view). However, the underlying disparities are not known in general. Hence the important task in intermediate view synthesis is to obtain suitable disparity estimates for the intermediate view pixels, in the presence of occlusions and erroneous left-right disparity estimates.

1. This discussion implicitly assumes that disparity estimates with infinite horizontal resolution are available. However, in practice, only half-pixel accurate disparity estimates are available.



w_{fg} - width of an occluding object

δ_{fg} - Disparity of the occluding object

E^r_{occ} - extent of right occlusion

δ^r_{occ}- Disparity of the right occlusion

E^l_{occ} - extent of left occlusion

δ^l_{occ}- Disparity of the left occlusion

FIG. 5.3: Occluded regions in the left, right, and intermediate views

5.3.1 IVS algorithm

We develop a synthesis scheme that would enable us to evaluate the suitability of the disparity maps obtained using DBS and FBS-based DCP. The steps of the algorithm for each scan-line are:

Step 1:

Proceeding from left to right, the intermediate view pixel corresponding to each left view pixel is computed using (EQ 5.3) and rounded to the nearest integer.

The left view pixel intensity and the scaled disparity are stored in two arrays at the computed intermediate view pixel location.

With the left-to-right scan, if a pixel has multiple candidate matches, then the match corresponding to the smallest depth[1] would overwrite the rest of the matches.

Due to regions visible only in the right view and due to disparity estimation errors, some of the intermediate view pixels will have no predictions.

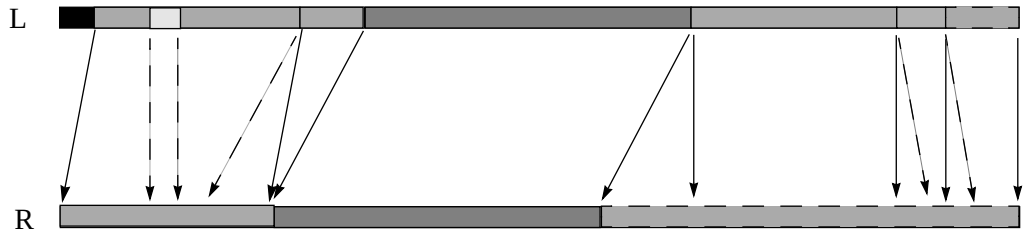
Step 2:

The groups of pixels with no available prediction are identified.

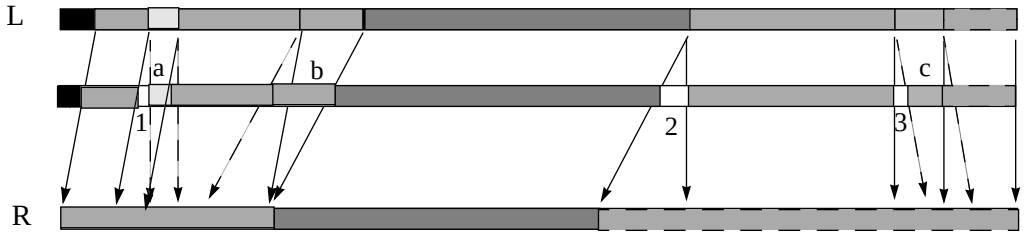
If the depth of the pixels to the left of a group is less than the depth of the pixels to its right, then the disparity value to the left of the group is used as the prediction for the group and the corresponding intensity values are predicted from the left view using (EQ 5.3).

If not, the disparity value to the right of the group is used as the prediction for the group and the

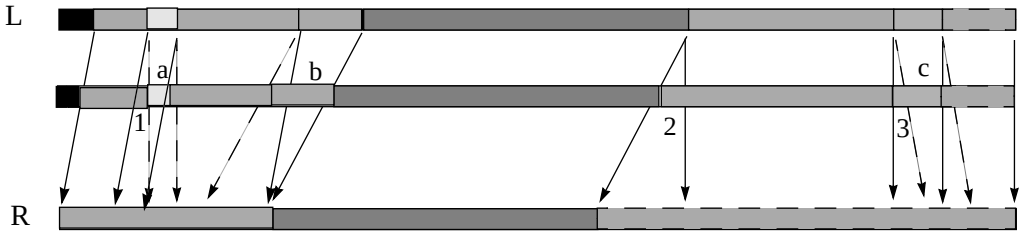
1. Disparity can take positive and negative values if the image sensors are shifted. But the disparity corresponds to an actual depth which is always positive. In our discussion, we assume that the sensors have been so shifted that all disparities are positive.



Groups of pixels on the left view scan-line with their estimated disparities are shown. The true estimates are shown as dark lines and the erroneous estimates are shown in dashed lines. The intensity of the pixels in the left view with wrong estimates are shown using a different shade for clarity. The actual intensities are shown in the right view.
Note that the region to the left of the occluding region has a false disparity estimate.



Illustrates step-1 of the IVS algorithm. The effect of three different types of errors in estimation are shown. The groups labeled a, b, and c correspond to false disparity estimates. Three groups of pixels without prediction (numbered 1, 2, and 3) can be identified. Only group-2 is due to occlusion. The other two are due to estimation errors.



Illustrates step-2 of the IVS algorithm where groups 1, 2, and 3 are filled. Groups 1 and 2 are filled correctly. But group-3 is filled incorrectly. Thus groups a, b, c, and 3 will have incorrectly synthesized values at the end of the synthesis.

FIG. 5.4: Illustration of the intermediate view synthesis algorithm in Section 5.3.1

corresponding intensity values are predicted from the right view using (EQ 5.3).

If the group is at the right extreme of the scan-line, the estimates to the left of the group are extended and the intensities are predicted from the right view using (EQ 5.3).

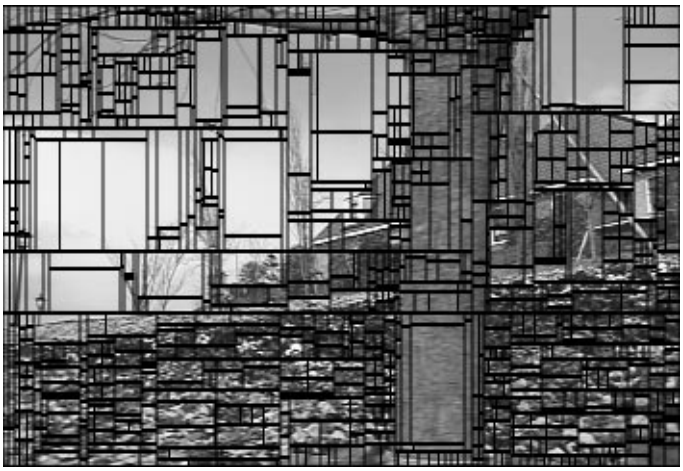
Such prediction is based on the assumption that the occluded region is a part of the nearest unoccluded background region.

The steps of the algorithm are illustrated in Fig. 5.4 for some common (but by no means exhaustive) scenarios. That this simple algorithm cannot handle errors in estimation in a correct fashion can be seen from Fig. 5.4. Incorrect matches can be detected using several methods. One commonly used method is to check for two-way consistency. A best match for a block in the left view, say B_L, is computed from the right view; for this best matching block in the right view B_R, a best match is computed in the left view; if this best match is not the same as B_L, the prediction for B_L is ignored during intermediate view synthesis. During matching, the presence of large errors over the entire search range or the absence of large variations in the distortion over the search range can also be used to identify potential occlusions and spurious matches. However, the above methods require transmitting an additional bit per block to represent the confidence in the estimated disparity; these methods are not considered here.

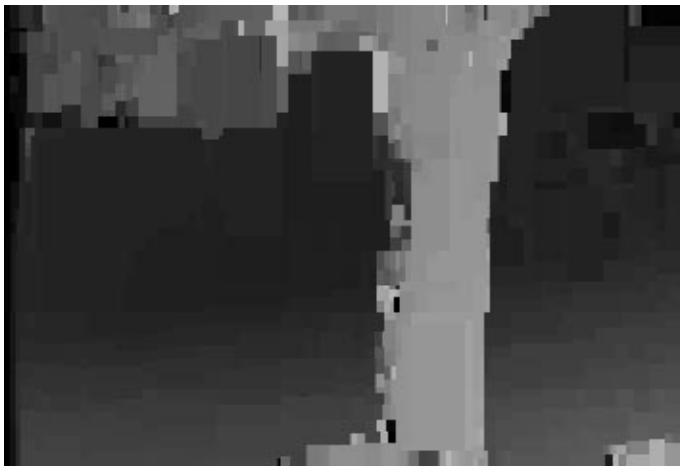
5.3.2 Evaluation of synthesized intermediate views

Based on the disparity maps obtained using the disparity-based segmentation algorithm of Chapter 3 and an 8x8 fixed block-sized based disparity compensation method, intermediate views were synthesized using the above algorithm for three different stereopairs. These stereo-pairs are referred to as *flower-garden*, *lab1* and *lab2*. Appendix A provides a description of these stereopairs and how they were obtained.

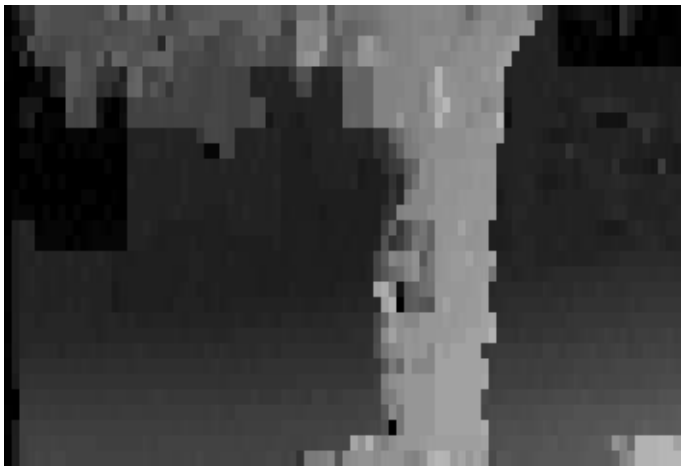
The intermediate view corresponding to 35% of the left-right baseline for the flower-garden stereopair was synthesized using the two methods. The respective synthesized views are shown in Fig. 5.5, along with the disparity maps and the coding details. It can be seen that the DBS results in a better synthesis that preserves disparity discontinuities and causes fewer artifacts when compared to FBS-DCP. However, the region to the left of the tree trunk, which is visible in the left view but occluded in the right view, is not well defined due to the use of the erroneous estimates during IVS; some of these regions have the same disparity estimate as the tree trunk; hence in addition to having a false prediction in their actual location, they also occlude the correct predictions to their immediate left. Since the position of the intermediate views are not known for this stereopair, we cannot compare the synthesized views with an actual view. For this reason, the *lab1* and *lab2* multi-view sets were created with known baseline distances. The synthesis performance for these two stereopairs are shown in Fig. 5.6 and Fig. 5.7, respectively. From the error images w.r.t the actual intermediate views, we can see that the synthesis procedure does in



(a) Disparity based segmentation
(1131 blocks, PSNR = 25.03 dB, 0.095 bpp)



(b) Disparity map using DBS



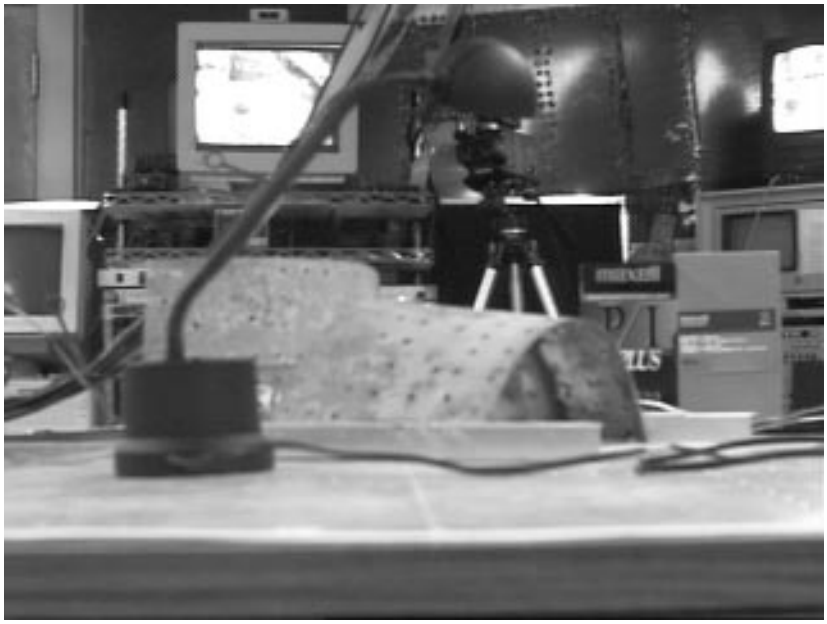
(c) Disparity map using FBS-DCP
(2640 blocks, PSNR = 24.67 dB, 0.102 bpp)

(d) Intermediate view (α=0.35) synthesized using the DBS disparity map

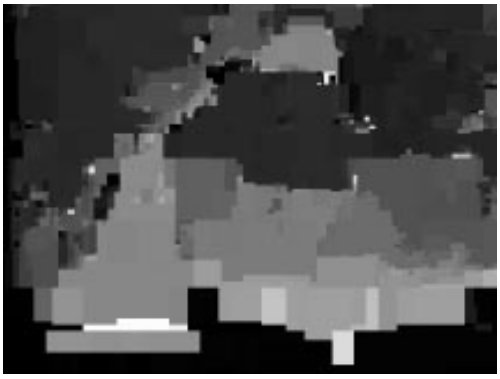
(e) Intermediate view (α=0.35) synthesized using the FBS-DCP disparity map

FIG. 5.5: (a)-(e) Comparison of intermediate views synthesized using disparity maps from DBS and FBS-DCP for the *flower-garden* stereopair

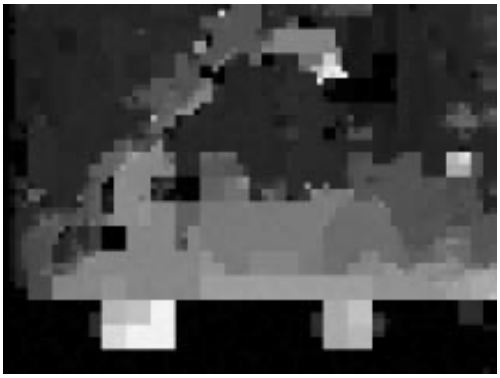
The disparity map from FBS-DCP causes incorrect foreground edges and the spurious matches result in synthesis errors, compared to the disparity map from DBS. The continuity in some flower branches is broken and holes appear in the tree trunk in (e).



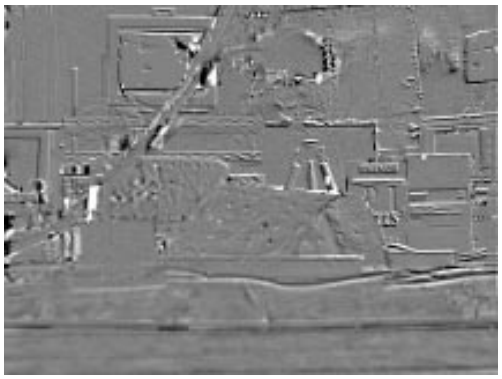
(a) The actual intermediate view corresponding to $\alpha = 0.5$



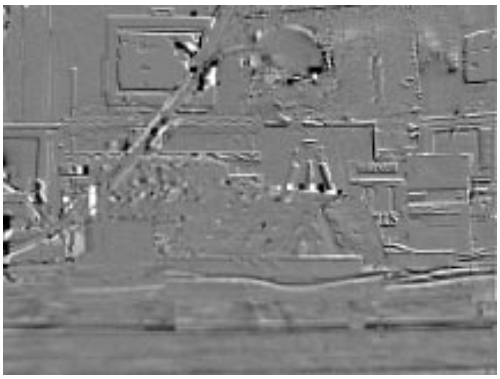
(b) Disparity map using DBS
(1547 blocks, PSNR = 28.79 dB, 0.077 bpp)



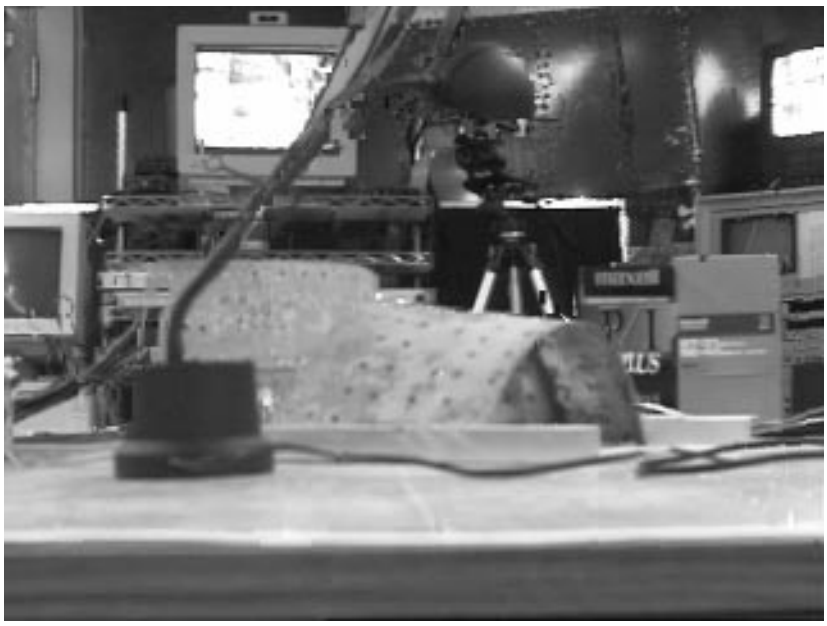
(c) Disparity map using FBS-DCP
(4800 blocks, PSNR = 28.87 dB, 0.128 bpp)



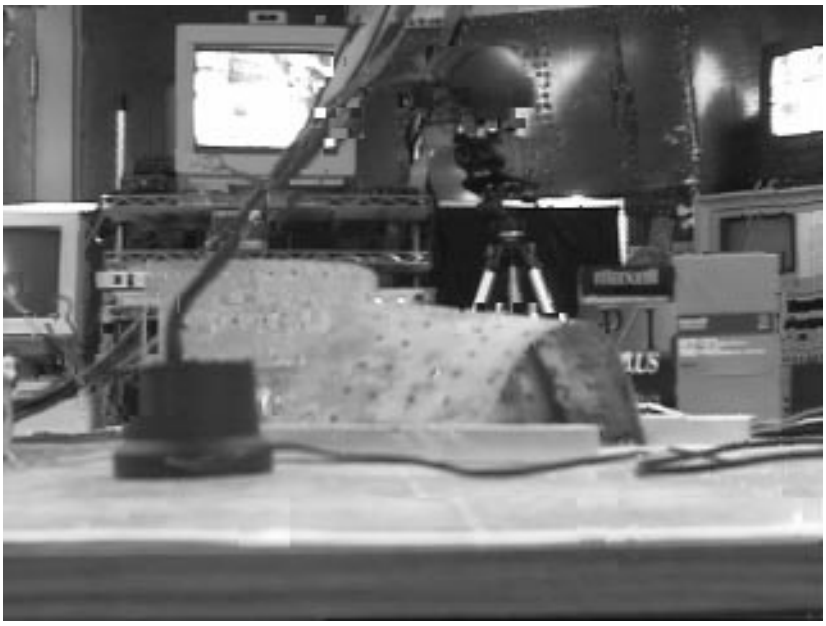
(d) Error in the synthesized intermediate view using the DBS disparity map (PSNR=23.6dB) [(a) - (f)]



(e) Error in the synthesized intermediate view using FBS-DCP disparity map (PSNR=23.3dB) [(a) - (g)]



(f) Intermediate view (for α=0.5) synthesized using the DBS disparity map

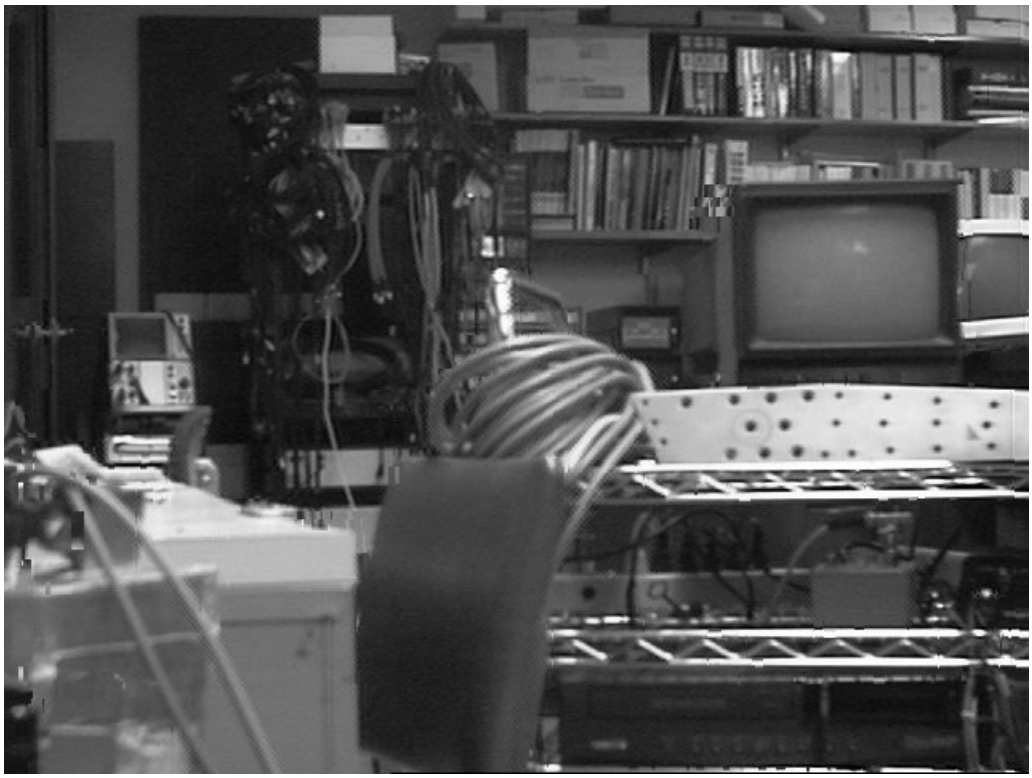


(g) Intermediate view (for α=0.5) synthesized using the FBS-DCP disparity map

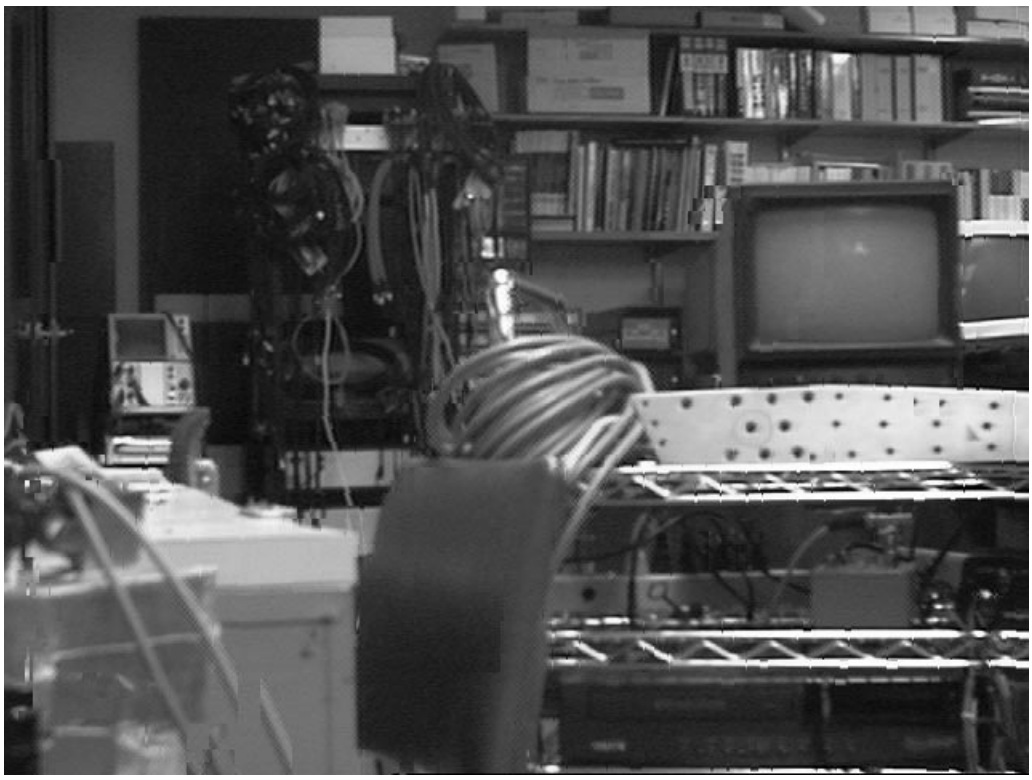
FIG. 5.6: (a)-(g) Comparison of intermediate views synthesized using disparity maps from DBS and FBS-DCP for the *lab1* multi-view set

The baseline between the left and right cameras is 4 cm. The PSNR in (d) and (e) are low due to possible errors in the baseline measurement. The errors are displayed (in d and e) to demonstrate that the IVS is reasonably accurate. It can be seen by comparing (f) and (g) that the spurious matches in FBS-DCP result in more object distortions in the synthesized view.

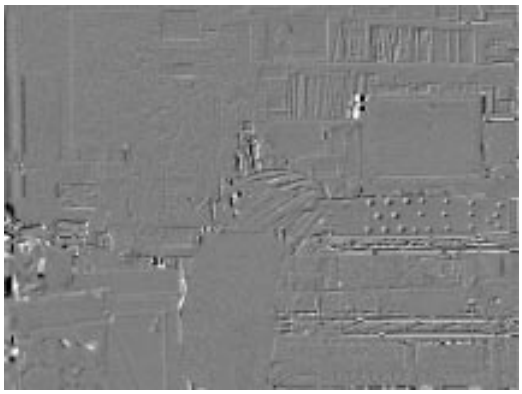
(a) The actual intermediate view corresponding to $\alpha = 0.5$



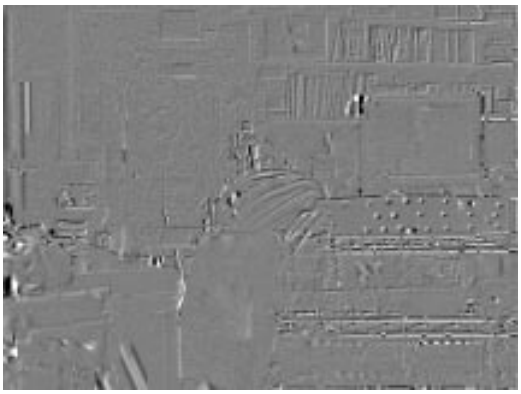
(b) Intermediate view (for $\alpha=0.5$) synthesized using the DBS disparity map



(c) Intermediate view (for α=0.5) synthesized using the FBS-DCP disparity map



(d) Error in the synthesized intermediate view using the DBS disparity map (PSNR=26.92 dB) [(a)-(b)]



(e) Error in the synthesized intermediate view using the FBS-DCP disparity map (PSNR=26.67 dB) [(a)-(c)]

FIG. 5.7: (a)-(e) Comparison of intermediate views synthesized using disparity maps from DBS and FBS-DCP for the *lab2* multi-view set

The baseline between the left and right cameras is 2 cm. DBS results in 1226 blocks, provides a DCP PSNR of 27.99 dB and requires 0.058 bpp to code the disparities. FBS-DCP has 4800 blocks, provides a DCP PSNR of 27.89 dB and requires 0.109 bpp to code the disparities. Compared to (b), (c) has more visually detectable errors (such as the cord in the front)

fact predict a reasonably accurate intermediate view. For these two stereopairs also, the disparity map from DBS results in fewer noticeable distortions in the synthesized intermediate views.

The better perceived quality of the synthesized views in the DBS case can be attributed to the better quality of the disparity map, which preserves disparity discontinuities better and has fewer spurious matches in featureless areas. Hence, in addition to providing a more compact representation, the DBS algorithm also provides better quality synthesized intermediate views.

5.4 MULTI-VIEW CODING EXTENSIONS

In this section we suggest possible multi-view extensions of the 2-view sequence compression methods described in the last chapter. We assume that an odd number of views ($2k$+1) are generated. The view from the central camera is coded at a higher quality for quality compatibility with monoscopic transmission; we call this view as the *main view.* The k views on each side of the main sequence will be referred to as *auxiliary views* as they will typically be coded at a lower quality. Since synthesis of intermediate views is essential to present a continuum of views, we do not consider configuration-2 coding schemes of Chapter 4.

5.4.1 Multi-view extensions of FBS-1 and DBS-1

The FBS-1 and DBS-1 schemes can be directly extended to code multiple views by coding each auxiliary view using disparity compensation w.r.t the corresponding main view frame. The regions uncompensated after DCP can be coded using intra-view motion compensation as described in Section 4.2. For the DBS-1 scheme, disparity-based segmentation has to be performed for each auxiliary frame in each auxiliary view.

As the baseline distance between an auxiliary view and the main view increases, the cross-correlation between them decreases. To handle this situation, we can modify the disparity compensation to depend on two reference frames, namely, the corresponding main sequence frame and the corresponding frame in the nearest auxiliary sequence that has already been coded. In such a scheme, unoccluded regions will be predicted from the higher quality main sequence frames, while the regions occluded in the main sequence frames can be predicted from the nearest auxiliary sequence frame. The disparities estimated w.r.t these two reference frames can be used for intermediate view synthesis. Such a prediction configuration is illustrated in Fig. 5.8.

5.4.2 Multi-view extension of RDBS

The DBS-1 extension in the previous subsection requires $2k$ segmentations per multi-view frame and all the $2k$ disparity maps need to be transmitted. Since the unoccluded regions have very high correlation across the views, if we can compute a single reliable disparity map using all the views, the regions unoccluded in all the views can be predicted using only this disparity map and an encoded single view. The prediction direction has to be reversed and the disparities have to be

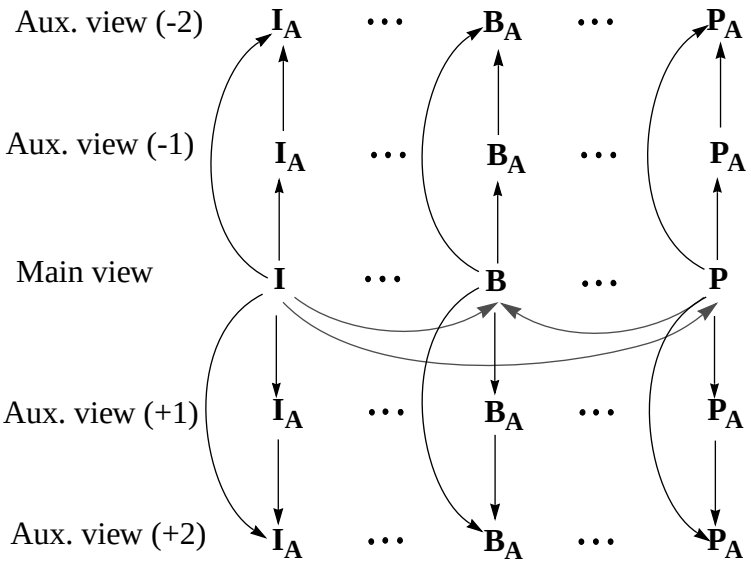


FIG. 5.8: Multi-view extension of FBS-1 and DBS-1 coding schemes

suitably scaled for the different baselines for such a prediction. The uncompensated regions in the different views can either be filled-in using the spatial prediction described in Section 4.5.2 or can be predicted from the nearest auxiliary view that has already been coded. Thus, the RDBS scheme can also be extended directly to multi-view prediction if the reliability of the disparity map can be improved by utilizing the information from the different views.

Multiple-baseline stereo matching

Multiple-baseline stereo matching is a well-known computer vision algorithm [119, 120] in which knowledge of the camera separations for a given multi-view set is used to obtain unambiguous disparity estimates. The principle behind the algorithm is as follows. During disparity estimation, the absence of features in a region, or, inherent ambiguities present in the scene (such as a periodic pattern) can lead to spurious matches. When multiple views obtained with different baselines are available, the distortion functions (such as mean absolute difference or mean of squared differences) for all the different baselines, computed for each block over a search range and normalized to a common baseline distance, will have a minimum at the location corresponding to the actual disparity of the block; the other minima due to ambiguous matches will not be aligned after such normalization. If these normalized distortion functions are summed, a unique minimum corresponding to the actual value can be obtained.

Hence by computing the disparity for the main view pixels w.r.t to auxiliary views using the multiple-baseline stereo matching algorithm, a single reliable disparity map can be obtained. A suitable adaptation of this algorithm for a 3-view case using the mean-absolute-difference criterion is presented in Section 5.5.1. Since only one segmentation per multi-view frame is performed, the computational complexity scales well with multiple views. As only one disparity map needs to be

transmitted, we speculate that better compression would be achieved compared to DBS-1 as the number of views increases.

5.4.3 Multi-view extension of ST-1

The ST-1 scheme can be extended for multi-view coding by modifying the multiple-baseline disparity based segmentation to be both motion and disparity adaptive. The segments thus obtained can be tracked over time and across views to obtain the predictions for the B, I_A, P_A and B_A frames. Only one segmentation per multi-view reference frame is required in this case. The coherence relationship can be used during tracking to reduce the search range.

5.5 3-VIEW CONFIGURATION

The performances of FBS-1, DBS-1, RDBS and ST-1 methods for a 3-view case can be extrapolated from the performances for the 2-view case presented in Chapter 4. This is because the 3-view configuration can be considered as a symmetric extension of a 2-view configuration. Since the baseline distances between the auxiliary views and the main view are equal, the overall excess bandwidth for the 3-view configuration will be very close to twice the excess bandwidth required in the 2-view case.

Even when the baseline between the main and auxiliary views is chosen to be equal to the nominal inter-ocular separation, intermediate views can be synthesized over a limited range to provide the correct perspectives during head movements over half the inter-ocular separation to the left and right of the central position.

Most people typically have a dominant eye that perceives the scene sharper than the other eye. In the 2-view case, the auxiliary view is arbitrarily chosen. This may affect the quality perceived by a viewer when the reduced quality view is presented to the viewer's dominant eye. When 3-views are coded with the main sequence at the center, depending on the viewer's preference the left and central views or the central and right views can be presented. In a broadcast situation over a channel such as a subscribed cable, the viewer's preference can be communicated using a low bandwidth reverse channel. In such a scenario, the content provider needs to transmit only the information needed to reconstruct two views.

In the following subsection, we present the modifications to disparity estimation needed to perform multiple-baseline stereo matching for the RDBS extension.

5.5.1 Multiple-baseline stereo matching for 3-views

In the RDBS extension, the disparity for each block in the main view is to be estimated w.r.t the two auxiliary views. Since the main view is the central view, the baseline distances to the two auxiliary views are $+b$ and $-b$, respectively. This implies that if the correct match for a block is at a

distance *d* in the right auxiliary view, then the distance to the correct match in the left auxiliary view is *-d*. However, if an incorrect minimum exists in the right view at *(d+ε)*, it would exist in the left view at *(-d+ε)*. This allows an easy modification of the block matching. The following function is evaluated for each block and the minimum of the function over the search range is chosen as the correct match.

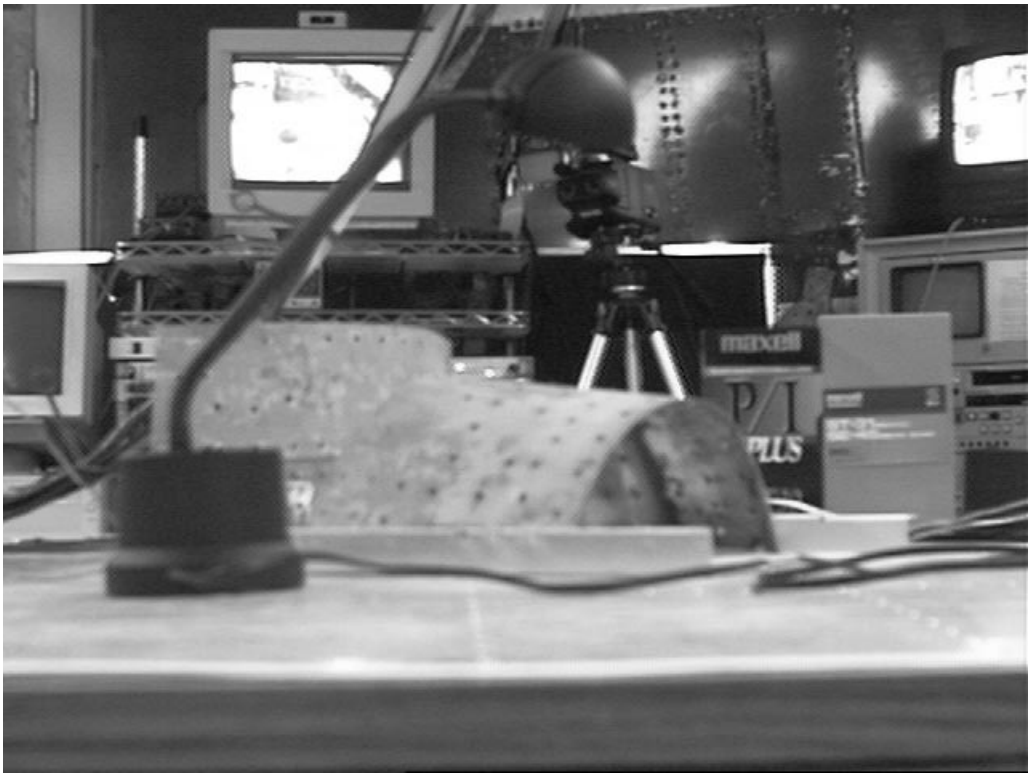
$$MAE(l) = \frac{1}{N_B}\left(\sum_{i,\, j \in B} \left|(I_C(i,j) - I_L(i,j-l))\right| + \sum_{i,\, j \in B} \left|(I_C(i,j) - I_R(i,j+l))\right| \right) \quad \text{(EQ 5.4)}$$

where I_L, I_C, I_R are used to represent the intensities of the left, central, and right images, B is the set of all pixels within the block, N_B is the number of pixels within the block and l is defined over the search range. The incorrect individual minima average out during the summation and the correct minimum is obtained.

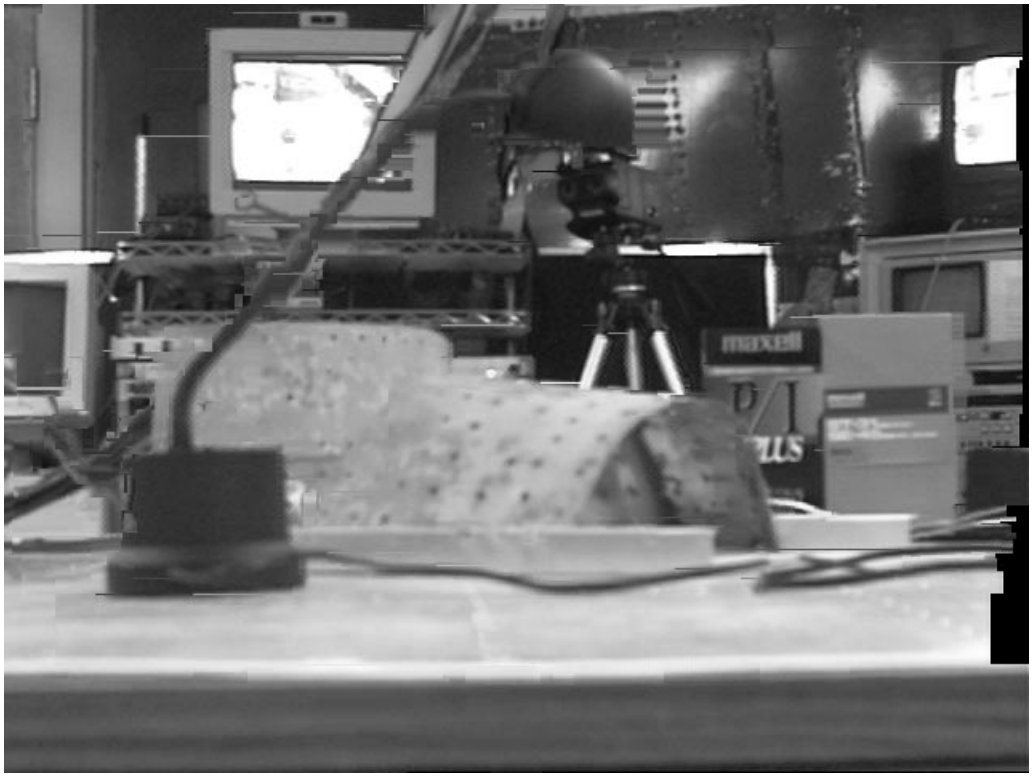
Using this modification, the disparity-based segmentation of the main view frame was performed using the *lab1* 3-view set described in Appendix A. Using the computed disparity map and reversing the direction of prediction the auxiliary views were predicted. The predicted right view, after filling-in, is shown in Fig. 5.9. It can be seen that a better prediction is achieved with two baselines than with just one baseline.

5.6 CONCLUSIONS

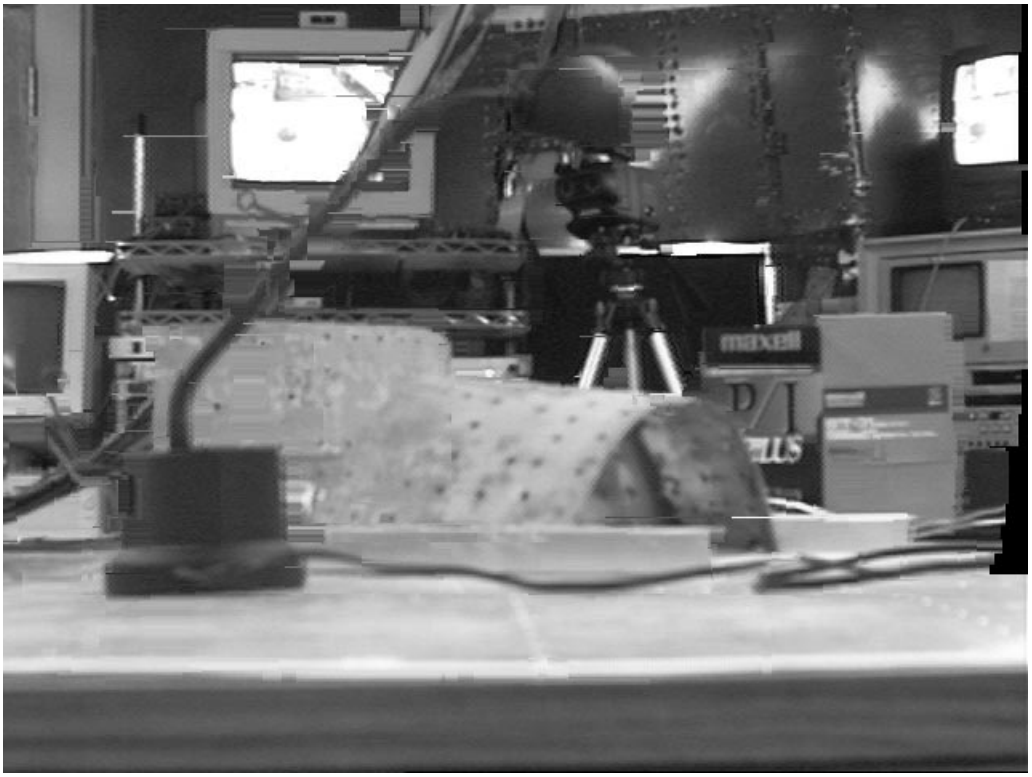
The issues in compression of multiple views and synthesis of intermediate views, which are needed to provide a viewer with motion parallax perception during head movements, were discussed in this chapter. Through a simple scheme for synthesizing intermediate views, we have demonstrated that the disparity map obtained during disparity-based segmentation is better suited for intermediate view synthesis than a disparity map obtained from a fixed block-size based disparity compensation method. We have outlined possible multi-view extensions to the stereoscopic sequence compression scheme described in Chapter 4. Additional experiments are required to evaluate the performance of such extensions. Potential advantages of a simple 3-view configuration, for which the performance of the coding methods can be extrapolated from their performance in the 2-view case, have been highlighted. Improvement in the accuracy of the disparity map by using multiple baselines has been demonstrated for a test case. Further experiments over a larger test set are needed to substantiate the coding improvements that can be achieved by using multiple-baseline stereo matching.



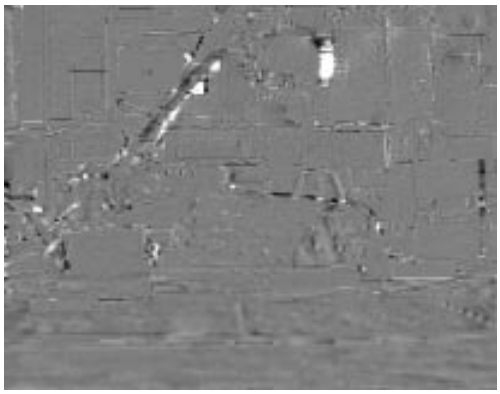
(a) The actual right view



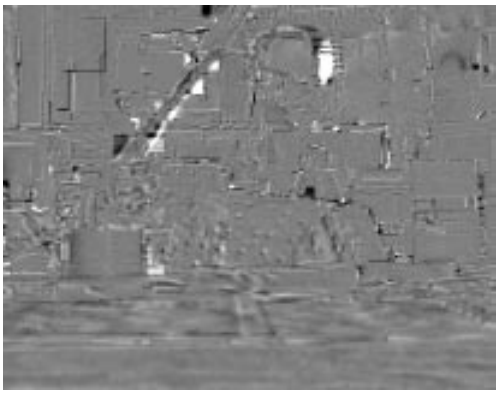
(b) Right view estimated using the multi-baseline disparity map



(c) Right view estimated using a single baseline disparity map



(d) Prediction error in the right view estimated using the multi-baseline disparity map (PSNR = 27.63 dB)



(e) Prediction error in the right view estimated using the single baseline disparity map (PSNR = 27.02 dB)

FIG. 5.9: (a)-(e) Comparison between the predicted auxiliary views obtained using a 2-baseline disparity map and a single baseline disparity map - *lab1* multi-view set

left-center baseline = center-right baseline = 2cm

It can be seen from (d) and (e) that a better compensation of the unoccluded regions is obtained with two baselines than with a single baseline.

Chapter 6

Conclusions and Future directions

In this chapter we summarize the key results presented in Chapters 3, 4, and 5, draw conclusions and emphasize the important contributions of this thesis. We also present possible future directions to improve the performance of our framework or to extend the framework to other related applications.

6.1 SUMMARY

In this thesis, we have presented a framework for coding stereoscopic image sequences that is well suited for a broadcast scenario. The issues affecting this problem and the solutions provided within our framework are summarized below:

- Quality compatibility with existing monoscopic transmission schemes is achieved by coding one of the sequences (the main sequence) at a similar quality.
- The excess bandwidth, over the bandwidth needed for single sequence transmission, is reduced by trading off the quality and/or resolution of the auxiliary sequences, so that it can be adjusted to be commensurate with the demand for stereoscopic video.
- The excess bandwidth is made scene-adaptive by using a disparity- and motion-adaptive quadtree-based segmentation technique.
- A reasonable stereoscopic video quality is achieved by accommodating the psychophysically motivated mixed-resolution based coding within the framework and by designing a residual coder that can suppress the most visually displeasing artifacts for a given coding rate.
- The need for a disparity map at the decoder for each stereoscopic frame to synthesize intermediate views is addressed by formulating a coding configuration, configuration-1, in which disparity compensation is used to predict each auxiliary sequence frame.
- Another configuration, configuration-2, is designed to increase compression by exploiting both inter-view and intra-view correlations.
- The quality of the synthesized intermediate view is improved by the use of an improved disparity map obtained through disparity-based segmentation.
- The coding scheme is made independent of the nature of the scene, and hence suitable for coding a wide variety of scenes, by using conventional waveform-based coding methods.

- Moderate encoder complexity and low decoder complexity are achieved by using simple block matching based motion/disparity estimation.
- Desirable features of prevalent single sequence compression standards, such as editability, random access, and independent decodability of a segment, are achieved for the stereoscopic sequence as well, by suitably extending the frame structure used in these standards.
- Computational complexity and excess bandwidth requirements for multi-view compression are reduced by the use of two joint coding schemes, namely, RDBS and ST-1.

Our novel disparity-based segmentation algorithm segments one view of a stereoscopic image pair based on the binocular disparity w.r.t the other view. Hence it adapts the number of disparity coding bits to the local disparity detail present within the stereopair. The segmentation is made computationally efficient by incorporating it within a multiresolution framework. A generalized quadtree decomposition is used for the segmentation to reduce the segmentation representation overhead. The partitioning locations for irregular QTD are computed using intensity edge information within a block. The segmentation overhead is further reduced by using regular partition at finer resolution levels, albeit at the expense of small inaccuracies in the disparity discontinuities. The conclusions, based on experiments over a wide variety of stereopairs, are:

- A 25-55% saving in bits needed to code the block disparities is achieved using DBS, when compared to fixed block-size based disparity compensation.
- The DBS also results in a disparity map that is more accurate, and hence better suited for synthesizing intermediate views, when compared to FBS-DCP.

Two configurations for coding stereoscopic image sequences are considered. In configuration-1, to facilitate synthesis of intermediate views at the decoder, a disparity map is transmitted for each stereo-frame. In configuration-2, the intermediate view synthesis capability is traded for increased compression efficiency by performing joint intra- and inter-view prediction of auxiliary frames. In addition to one FBS-based and one MR-QTD based dependent coding extension for each configuration, namely, FBS-1, DBS-1, FBS-2 and DBS-2, two joint coding extensions that belong to configuration-1 namely, RDBS and ST-1, have been presented in Chapter 4. A quadtree and vector/scalar quantizer based residual coder suited for low bit-rate residual coding is used to control the coded quality of the stereoscopic sequence. The resulting rate-distortion performances of the different extensions, over six stereoscopic test sequences, have been used to compare the performance of MR-QTD based extensions against the performance of FBS-based extensions in each configuration. The conclusions based on these experiments are:

- MR-QTD based extensions consistently outperform the FBS-based extensions. The MR-QTD methods offer an improvement in the PSNR of about 1-2 dB over FBS-based methods at a given low excess bandwidth. The percentage improvement is higher at lower excess

bandwidths. Hence, the MR-QTD methods will have a definite edge over FBS-based extensions at low excess bandwidths.

- Configuration-2 schemes have better R-D performance than corresponding configuration-1 schemes. Hence, if synthesis of intermediate views is not needed at the decoder, this configuration should be preferred over configuration-1. The quality of the reference frames becomes important in this case. For instance, if the auxiliary sequences are coded at very low excess bandwidths, inter-view prediction w.r.t the good quality main sequence frames will be preferred to intra-view prediction. In such a case, since a large fraction of the disparity map is transmitted anyway, to facilitate synthesis of intermediate views the rest of the disparities can also be coded.

- The RDBS method has slightly better performance than DBS-1 for both the main and auxiliary sequences in most of the test cases. This method is a good candidate for multi-view compression with intermediate view synthesis, as the segmentation complexity and coding efficiency scale well with multiple views. With multiple-baseline disparity estimation, the coding performance will be significantly better than a multi-view extension of DBS-1.

- The ST-1 scheme requires a higher coding rate than DBS-1 for the main sequence. This is due to the increase in residual coding overhead at the high quality required for the main sequence. However, this scheme has the advantage that the B- and B_A-frames need not be subjected to MR decomposition, as the disparity and motion search are simplified by the use of the coherence relationship between disparity and motion between two stereo-frames. Hence this extension can be used in applications that require low computational complexity while allowing lower quality levels for the main sequence.

- Mixed-resolution based coding can be used to further decrease the excess bandwidth in all cases, while reasonably preserving the perceived stereoscopic quality.

The methods are compared using a single objective measure, the PSNR. Possible perceptual improvements due to segmentation have not been quantified. Subjective tests over a wide cross-section of viewers are needed to evaluate the perceived stereoscopic quality.

6.2 FUTURE DIRECTIONS

Even at low excess bandwidths, a significant portion of bit-budget is spent on coding the residuals. If the disparity or motion compensation can be improved, the residual coding overhead can be considerably reduced. We have considered only simple block matching. As a first step to improve compensation, overlapped block motion compensation [27] can be applied to obtain interpolated motion and disparity estimates for each pixel within a block. By relaxing the

‘translation of planar patch parallel to the image sensor’ assumption for motion and disparity and considering affine or perspective transformation based models [6] (see Section 2.2.5), the compensation can be further improved.

The segmentation boundaries in our method are computed based only on the luminance components. By suitably exploiting the chrominance components also, the segmentation can be improved in regions of near iso-luminance.

The knowledge of depth, obtained by using multiple cameras, can be used in conjunction with other a priori or derived knowledge about the image to generate simpler and more compact representations. For example, depth-adaptive bit allocation can be carried out if the depth range of interest in the scene is known. This has been used to detect background in typical ‘head and shoulders’ type videophone sequences in [39]. The segmented background region is coded at a lower quality. Similar extensions for more general scenes deserve further consideration.

A significant number of bits are wasted filling-in the uncovered regions in the segment tracking case. If the background region over a group of frames can be extracted (for e.g. using mosaic-based coding techniques [81, 115]) and coded as a single image, then the foreground region can be overlaid. This can give rise to perceptually pleasing images even at low bit-rates.

A perceptually adaptive residual coding scheme that takes into account stereoscopic masking effects, such as suppression of artifacts in a coarsely quantized view when the corresponding regions in the other view are coded at a higher quality, is presented in [101]. A similar extension within our framework can lead to further reduction in the excess bandwidth without affecting the perceived stereoscopic quality.

The extension of the framework to multiple views also requires further consideration. Information from psychophysical experiments coupled with better IVS capability have to be considered to decide on a reasonable camera spacing and geometry in a multi-view case.

Though our framework will be able to handle small scale changes in objects over time due to ‘camera zoom’, we have not explicitly considered this problem. When the focal lengths of two cameras with a fixed baseline distance are changed, the resulting views when viewed stereoscopically would appear either compressed or stretched in depth due to the non-unity angular magnification. Further consideration based on subjective experiments is required to arrive at a reasonable mechanism to correct for this by suitably varying the baseline separation along with the focal length.

Appendix A
Description of stereoscopic test images and sequences

In this appendix, we present brief descriptions about the different stereoscopic test images and sequences used in this thesis.

The *booksale* and *crowd* sequences were created by us using a field-sequential (left and right views are recorded as odd and even fields of a frame), fixed-focus (9.5mm) stereoscopic camera (from Toshiba) with an inter-camera separation of 50mm. The cameras were aligned to be parallel by us. The NTSC resolution video signal was digitized to 640 x 240 pixels/field. The frame rate is 30 frames-per-second. Because of the field interlace, the left and right eye views are not acquired at the same time and are offset by 1/60th of a second. Hence, the disparity vectors estimated using successive left and right fields would have motion components in addition to the actual disparity. In this thesis, we have reported results that use 81 frames of the *booksale* sequence and 96 frames of the *crowd* sequence. The *booksale* sequence has a panning movement of the cameras, small object displacements and a good sense of depth. The *crowd* sequence has small random camera and object displacements and a good depth range.

The *aqua*, *piano*, *train* and *tunnel* sequences were created within the DISTIMA project in Europe and provided by CCETT, France. The sequences were digitized from a PAL resolution video signal to 720x576 pixels/frame at 25 frames-per-second. The frames are composed of interlaced fields. These sequences were created using a fixed-focus stereoscopic camera with a focal length of 40mm and a camera baseline of 8.75cm. The cameras are aligned such that the vertical disparity between the left and right frames is less than 1 pixel. (The convergence distance for the cameras are 2.35 m, 5.2 m, 2.8 m, and 2.8 m, respectively, for the four sequences). Except for the *tunnel* sequence which has 100 frames, the rest of the sequences have only 50 frames each. The *aqua* sequence has very high spatial detail, but very little object and camera motions. The *piano* sequence has a mixture of plain and highly textured regions with nonrigid displacements and the camera pans the scene. The left and right cameras are not matched resulting in a significant difference in the luminance and chrominance components. The *train* sequence has regions with varying degrees of displacements with significant occlusions between two passing train, one moving faster than the other. There is a slight panning movement of the camera as well. In the *tunnel* sequence, a train moves along a curve thus exhibiting a deformation in the observed shape

from frame to frame. The moving train also occludes and exposes objects farther from the cameras. Significant local illumination causes strong shadows of the train. There exists a certain color mismatch between the two views. A sample stereoscopic pair of frames from each of the six sequences are shown below (with lines doubled to maintain aspect ratio).

Left View Right View

A stereoscopic pair of frames from *booksale* sequence

A stereoscopic pair of frames from *crowd* sequence

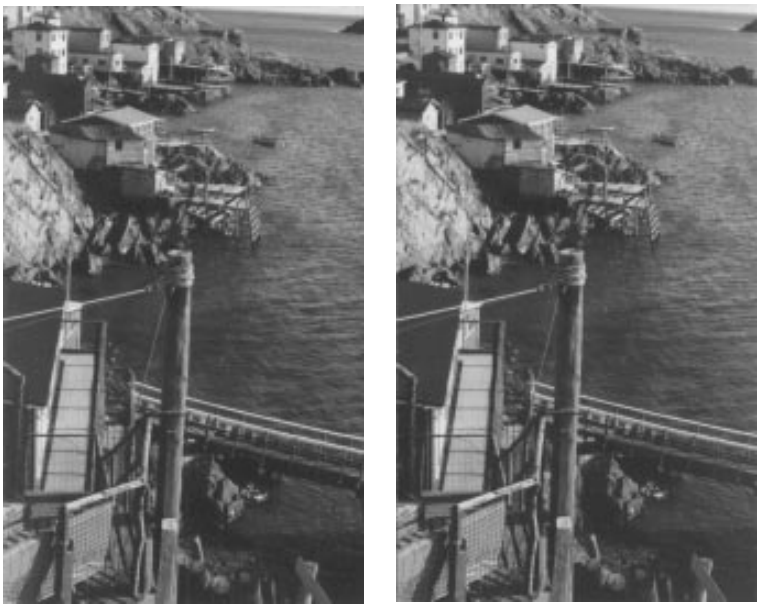
A stereoscopic pair of frames from *aqua* sequence

A stereoscopic pair of frames from *piano* sequence

A stereoscopic pair of frames from *train* sequence

A stereoscopic pair of frames from *tunnel* sequence

The odd fields of the first and fourth frames of the *flower garden* sequence shown as a stereopair
(Lines doubled to show with the correct aspect ratio)



Left and right views of the *lake* stereopair

Left and right views of the *group-photo* stereopair

In addition to using these sequences, we also use several individual stereoscopic image pairs and two multi-view image sets in Chapters 3 and 5. The *flower garden* sequence (704x240 pixels) is a test sequence widely used for testing monoscopic sequence compression methods. There is no object motion within the scene and the camera moves approximately along a line. For this reason, we can create stereo pairs from this sequence by using frames that are separated in time. The tree at the foreground, and the gradually receding ground, coupled with the high spatial detail in the foreground makes this a good test pair for judging our segmentation algorithm. The manner in which the *lake* (544x328 pixels) and *group-photo* (528x432 pixels) stereo pairs were acquired is not available.

The multiview sets, *lab1* and *lab2*, used in Chapter 5 were created by us with a camera (with f = 12.5mm) that was slid along a graduated rail. For the *lab1* set, the left-to-center view and the

Left view

Left view

Central view

Central view

Right view

lab1 3-view set

Right view

lab2 3-view set

center-to-right view baseline distances are 2 cm each. For the *lab2* set, the left-to-center view and the center-to-right view baseline distances are 1 cm each. The NTSC analog video was digitized to 640x480 pixels/frame. Smaller baselines than the nominal human inter-ocular separation are used because the objects in this case are quite close to the cameras; due to the non-unity angular magnification during viewing, the views with the nominal interocular separation are hard to fuse.

Appendix B
Motion and disparity vector coding

The motion and disparity vectors in the experiments were coded in a manner similar to the MPEG-2 test-model 5 (TM5) [26]. For the fixed block-size case, the block motion/disparity vectors are scanned left to right and from top to bottom. The horizontal and vertical components are coded separately. The first motion/disparity vector is coded by itself. The subsequent vector components are coded by first subtracting them from the previously coded component and then entropy coding the difference. Since the search range is usually scene-dependent, the MPEG-2 TM5 uses the following procedure to facilitate the use of fixed Huffman tables, which we also follow.

A symmetric search range is assumed. If the search range is $\pm k$ pixels, then $m = log_2(\lceil k \rceil)$ bits are needed to represent the magnitude of one component of the motion/disparity vector. After differencing, the range doubles, and hence $(m+1)$ bits are needed. To code half-pixel accurate estimates, an additional bit is needed. If the absolute value is non-zero, the sign is coded using 1 bit. Since the least significant bits of the magnitude of the differences are likely to be random and uniformly distributed, the least significant (m-2) bits are coded using fixed length codes (FLCs). The most significant 4 bits are coded using a pre-determined set of variable length codes (VLCs) that is typically designed to take advantage of the fact that large values of the differences are less likely.

To code the block motion/disparity vector in the segmentation case, the blocks are scanned according to the depth-first traversal of the quadtree. To code these vectors, the same procedure as described in the previous paragraph is used. For the RDBS and ST-1 schemes, where the direction of prediction is reversed, the half-pixel estimate is coded separately as discussed in Sections 4.5.2 and 4.5.3.

References

Books/compilations:

[1] Majid Rabbani and Paul W. Jones, “Digital image compression techniques”, SPIE Optical Engineering Press, Bellingham, WA, 1991.

[2] Arun N. Netravali and Barry G. Haskell, “Digital pictures: representation and compression”, Plenum Press, New York, 1988.

[3] Arun N. Netravali and Birendra Prasada, Editors, “Visual communication systems”, IEEE Press, New York, 1989.

[4] Russell Hsing and Andrew G. Tescher, Editors, “Selected papers on visual communication: technology and applications”, SPIE Optical Engineering Press, Bellingham, MA, 1990.

[5] Allen Gersho and Robert M. Gray, “Vector quantization and signal compression”, Kluwer Academic Publishers, Boston, 1992.

[6] P. Anandan, *et al*., “Hierarchical model-based motion estimation”, in “Motion Analysis and Image sequence processing”, Ibrahim Sezan and Reginald Lagendijk, Editors, Kluwer Academic Publishers, 1993.

[7] K. R. Rao and P. Yip, “Discrete cosine transform: algorithms, advantages, applications”, Academic Press, Boston, 1990.

[8] R. J. Clarke, “Transform coding of images”, Academic Press, Orlando, 1985.

[9] Anil K. Jain, “Fundamentals of digital image processing”, Prentice Hall, Englewood Cliffs, NJ, 1989.

[10] Bernd Jahne, “Digital Image Processing: Concepts, algorithms and scientific applications”, Springer-Verlag, 1993.

[11] P. P. Vaidyanathan, “Multirate systems and filter banks”, Prentice Hall, Englewood Cliffs, NJ, 1993.

[12] Bela Julesz, “Foundations of Cyclopean perception”, Univ. of Chicago Press, Chicago, 1971.

[13] David Marr, “Vision: a computational investigation into the human representation and processing of visual information”, W.H.Freeman and Co., San Francisco, 1982.

[14] Lenny Lipton, “Foundations of the stereoscopic cinema: a study in depth”, Van Nostrand Reinhold, New York, 1982.

[15] V. M. Bove, “Scalable (extensible, interoperable) digital video representation”, Chapter 3, Andrew B. Watson, Editor, “Digital images and human vision”, MIT Press, Cambridge, MA, 1993.

[16] John E. W. Mayhew and John P. Frisby, Editors, “3D model recognition from stereoscopic cues”, MIT Press, Cambridge, MA, 1991.

[17] A. Rosenfeld, Editor, Multiresolution image processing and analysis”, Springer-Verlag, New York 1984.

[18] R. O.Duda and P. E. Hart, “Pattern classification and scene analysis”, Wiley, NewYork, 1973.

[19] H. Samet, “Design and analysis of spatial data structures”, Addison-Wesley, Reading, MA, 1989.

[20] G. Wolberg, “Digital image warping”, IEEE Computer Society, Washington, 1990.

[21] D. F. McAllister, “Stereo computer graphics and other true 3D technologies”, Princeton University Press, Princeton NJ, October 1993..

Standards:

[22] “Video codec for audiovisual services at px64 kbits/s”, CCITT Recommendation H.261, Dec. 1989.

[23] “Description of Reference Model 8 (RM8)”, CCITT Doc. 525, June 1989.

[24] “Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s”, ISO 11172 (MPEG-1), Draft International Standard, November 1992.

[25] ISO/IEC JTC1/SC29/WG11, “Information technology - generic coding of moving pictures and associated audio”, Recommendation H.262, ISO/IEC 13818-2 (MPEG-2), Draft International Standard, March 1994.

[26] ISO/IEC JTC1/SC29/WG11, Test model editing committee, “MPEG-2 Video Test Model 5”, Doc. No. 400, April 1993.

[27] “Video coding for low bitrate communications”, Draft ITU-T Recommendation H. 263, December 1995.

[28] ISO/IEC JTC1/SC29/WG11, “MPEG4 proposal package description”, no. 937, “Test/ evaluation procedures document”, no. 938, “Requirements for the MPEG4 SDL”, no. 939, “Call for proposals”, no. 943, March 1995.

Journal/Transactions papers:

[29] F.Chassaing et.al, “A stereoscopic television system and compatible transmission on a MAC channel”, Signal Processing: Image Communication, no. 4, pp., 1991.

[30] Michael G. Perkins, “Data compression of stereopairs”, IEEE Trans. on Communications, vol.40, no. 4, pp. 684-696, April 1992.

[31] A. Tamtaoui and C. Labit, “Constrained disparity and motion estimators for 3DTV image sequence coding”, Signal Processing: Image Commun., vol. 4, pp. 45-54, 1991.

[32] R. Skerjanc and J.Liu, “A three camera approach for calculating disparity and synthesizing intermediate pictures”, Signal Processing: Image Commun., Vol.4, No.1, pp.55-64, 1991.

[33] J. Liu and R. Skerjanc, “Stereo and motion correspondence in a sequence of stereo images”, Signal Processing: Image Commun., Vol. 5, pp. 305-318, 1993.

[34] D. Tzovaras, M. G. Strintzis, and H. Sahinoglou, “Evaluation of multiresolution block matching techniques for motion and disparity estimation”, Signal Processing: Image Commun., vol. 6, no. 1, pp. 59-67, 1994.

[35] B. Kost and S. Pastoor, “Visibility thresholds for disparity quantization errors in stereoscopic displays”, Proc. of SID, vol. 32, no.2, pp. 165-170, 1991.

[36] Stephane G. Mallat, “A theory for multiresolution signal decomposition: The wavelet representation”, IEEE Trans. on PAMI, vol. 11, no. 7, pp. 674-693, July 1989.

[37] S. G. Mallat, “Multifrequency channel decompositions of image and wavelet models”, IEEE Trans, on ASSP, Vol. 37, no. 12, Dec. 1989.

[38] Ingrid Daubechies, “Orthonormal bases of compactly supported wavelets”, Communications of Pure and Applied Mathematics, vol. 41, pp. 909-996, 1988.

[39] M. Waldowski, “A new segmentation algorithm for videophone applications based on stereoscopic image pairs”, IEEE Trans. on Communications, Vol. 39, no. 12, pp. 1856-1868, December 1991.

[40] I. Grammalidis, *et al*., “stereoscopic image sequence coding based on three dimensional motion estimation and compensation”, Signal Processing: Image Commun., vol. 7, no. 2, pp. 129-145, 1995.

[41] X. Wu and Y. Fang, “A segmentation-based predictive multiresolution image coder”, IEEE Trans. on Image Processing, vol. 4, pp. 34-47, January 1995.

[42] X.Zhang, M. C. Cavenor and J. F. Arnold, “Adaptive quadtree coding of motion-compensated image sequences for use on the broadband ISDN”, IEEE Trans. on Circuits and Systems for Video Technology, vol.3, no.3, pp. 222-229, June 1993.

[43] Y. Linde, A. Buzo, R. Gray, “An algorithm for vector quantization design”, IEEE Trans. on Communications, COM-28, no. 1, pp. 84-95, January 1980.

[44] Stuart P. Lloyd, “Least squares quantization in PCM”, IEEE Trans. on Information Theory, Vol. IT-28, No. 2, pp. 129-134, March 1982.

[45] A. Asif and J. M. F. Moura, “Image codec by noncausal prediction, residual mean removal, and cascaded VQ”, IEEE Trans. on Circuits and Systems for Video Tech., Vol. 6, No. 1, pp. 42-55, February 1996.

[46] N. Balram and J. M. F. Moura, “Noncausal predictive image codec”, IEEE Trans. on Image Proc., Vol. 5, no.8, TBD, August 1996.

[47] B. G. Lee, “A new algorithm to compute the discrete cosine transform”, IEEE Trans. on ASSP, ASSP-32 (6), pp. 1243-1245, 1984.

[48] J. W. Woods and S. D. O’Neil, “Subband coding of images”, IEEE Trans. on ASSP, ASSP-34(5), pp.1278-1288, 1986.

[49] H. S. Malvar and D. H. Staelin, “The LOT: transform coding without blocking effects”, IEEE Trans. on ASSP-37 (4), pp. 553-559, April 1989.

[50] A. E. Jacquin, “Fractal image coding - a review”, Proceeding of the IEEE, vol.81, no.10, Oct. 1993.

[51] H. Li, *et al*., “Image sequence coding at very low bitrates: A review”, IEEE Trans. on image processing, Vol.3, no. 5, pp. 589-609, Sept. 1994.

[52] J. R. Jain and A. K. Jain, “Displacement measurement and its application in interframe image coding”, IEEE Trans. on Commun., vol.29, pp.1799-1806, Dec. 1981.

[53] H. G. Musmann, *et al*., “Advances in picture coding”, Proc. of the IEEE, Vol. 73, no. 4, pp. 523-548, Apr. 1985.

[54] B.Liu and A.Zaccarin, `New fast algorithms for the estimation of block motion vectors’, IEEE Trans. on Circuits and Systems, Vol.3, No.2, pp. 148-157, April 1993.

[55] H. G. Musmann, *et al*., “Object-oriented analysis synthesis coding of moving images”, Signal Processing: Image Commun., Vol. 1, no. 2, pp.117-138, October 1989.

[56] M. Hotter, “Object-oriented analysis-synthesis coding based on moving two dimensional objects”, Signal Processing: Image Commun., vol. 2, no. 4, pp. 409-428, Dec. 1990.

[57] M. Hotter, “Optimization and efficiency of an object-oriented analysis-synthesis coder”, IEEE Trans. on circuits and systems for video tech., Vol. 4, no.2, pp. 181-194, April 1994.

[58] K. M. Uz, M. Vetterli, D. J. LeGall, “Interpolative multiresolution coding of advanced television with compatible subchannels”, IEEE Trans. on circuits and systems for video

tech., Vol.1, No.1, pp. 86-99, March 1991.

[59] M. F. Chowdhury, *et al*., "A switched model-based coder for video signals", IEEE Trans. on circuits and systems for video tech., Vol. 4, no. 3, pp. 216-227, June 1994.

[60] C. S. Choi, *et al*., "Analysis and synthesis of facial image sequences in model-based image coding", IEEE Trans. on circuits and systems for video tech., Vol. 4, no. 3, pp. 257-275, June 1994.

[61] D. Taubman and A. Zakhor, "Multirate 3D subband coding of video", IEEE Trans. on image processing, vol. 3, no. 5, pp. 572-588, Sep. 1994.

[62] M. Kunt, et al., "Second generation image coding techniques", Proc. IEEE, vol. 73, pp. 549-574, April 1985.

[63] F. Meyer and S. Beucher, "Morphological segmentation", J. Visual Commun. & Image representation, vol. 1, no. 1, pp. 21-46, Sept. 1990.

[64] P. Salembier and M. Pardas, "Hierarchical morphological segmentation for image sequence coding", IEEE Trans. on image processing, vol. 3, no. 5, pp. 639-651, Sep. 1994.

[65] C-L. Huang and C-Y Hsu, "A new motion compensation method for image sequence coding using hierarchical grid interpolation", IEEE Trans. on circuits and systems for video tech., vol. 4, no. 1, pp. 42-52, Feb. 1994.

[66] T. R. Fischer, "A pyramid vector quantizer", IEEE Trans. on Inform. theory, vol. 32, no. 4, pp. 568-583, July 1986.

[67] H. S. Wang and N. Moayeri, "Trellis coded vector quantization", IEEE Trans. on Commun., vol. 40, no. 8, pp. 1273-1276, August 1992.

[68] M. Antonini, *et al.*, "Image coding using wavelet transform", IEEE Trans. on Image Proc., vol. 1, no. 2, pp. 205-220, April 1992.

[69] M. Barlaud, *et al.*, "Pyramidal lattice vector quantization for multiscale image coding", IEEE Trans. on Image Proc., vol. 3, no. 4, pp. 367-381, July 1994.

[70] P. C. Cosman, *et al.*, "Vector quantization of image subbands: a survey", IEEE Trans. on Image Proc., vol. 5, no. 2. pp. 202-225, Feb. 1996.

[71] J. M. Shapiro, "An embedded hierarchical image coder using zerotrees of wavelet coefficients", Proc. of Data Compression Conf., 1993, pp. 214-223.

[72] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design", IEEE Trans. on Signal Proc., vol. 40, no. 9, Sept. 1992.

[73] S. L. Horowitz and T. Pavlidis, "Picture segmentation by a tree traversal algorithm", J. Assoc. Computing Mach., vol. 23, pp. 368-388, 1976.

[74] T. Aach and A. Kaup, “Disparity-based segmentation of stereoscopic foreground/ background image sequences”, IEEE Trans. on Commun., vol. 42, no.2, pp. 673-679, Feb. 1994.

[75] R. M. Haralick and L. G. Shapiro, “Survey: Image segmentation techniques”, J. Computer vision, graphics and image proc., vol. 29, pp. 100-132, 1985.

[76] P.J.Burt and E.H.Adelson, ‘The Laplacian Pyramid as a compact image code', IEEE Trans. on Commun., Vol. 31, No. 4, pp. , April 1983.

[77] T. Okoshi, “Three-dimensional displays”, Proc. of the IEEE, vol. 68, no. 5, May 1980.

[78] L. Lipton, “The evolution of electronic stereoscopy”, SMPTE Journal, vol. 100, no. 5, pp. 332-336, May 1991.

[79] D. J. Le Gall, “MPEG: A video compression standard for multimedia applications”, Commun. of the ACM, vol. 34, no. 4, pp. 47-58,April 1991.

[80] M. Liou, “Overview of the px64 kbits/s video coding standard”, Commun. of the ACM, vol. 34, no. 4, pp. 60-63,,April 1991.

[81] M. Irani, et al., “Video compression using mosaic representations”, Signal Processing: Image Commun., vol. 7, no.4-6, pp.529-552, Nov. 1995.

[82] J. Y. A. Wang and E. H. Adelson, “Representing moving images with layers”, IEEE Trans. on Image Proc., vol. 3, no. 5, pp. 625-638, Sept. 1994.

[83] Y. Ohta and T. Kanade, “Stereo by intra- and inter- scanline search using dynamic programming”, IEEE Trans. PAMI, Vol. 7, no. 2, pp. 139-154, March 1985.

[84] C. E. Shannon, “Coding theorems for a discrete source with a fidelity criterion”, in IRE Nat. Conv. Rec., part4, pp.142-163, 1959.

[85] H. H. Baker and R. C. Bolles, ‘Generalizing epipolar-plane image analysis on the spatiotemporal surface’, Intl. J. of Computer Vision, Vol. 3, pp.33-49, 1989.

Conference/Workshop papers:

[86] Michael E. Lukacs, “Predictive coding of multi-viewpoint image sets”, in Proc. of ICASSP, 1986, pp. 521-524.

[87] A.Schertz, “Source coding of stereoscopic television pictures”, Third intl. conf. on image proc. and its applications, IEE Conf. Pub. no. 307, 1989, pp.462-464.

[88] A. Tamtaoui and C. Labit, “Coherent disparity and motion compensation in 3DTV image sequence coding schemes”, in Proc. of ICASSP, 1991, pp. 2845-2848.

[89] I. Dinstein, *et al.*, “Compression of stereoscopic images and the evaluation of its effects on

3D perception", in SPIE Conf. Applications of Digital Image Processing XII, 1989, Vol. 1153, pp. 522-530.

[90] J. Liu and R. Skerjanc, "Construction of intermediate pictures for a multiview 3D system", SPIE/IS&T Symp. on Electronic imaging, Vol. 1669, Feb. 1992, pp. 10-19.

[91] R. E. H. Franich, R. L. Lagendijk, and J. Biemond, "Stereo-enhanced displacement estimation by genetic block matching", in SPIE Conf. Visual Commun. Image Processing, 1993, Vol. 2094, pp. 362-371.

[92] T. Ozkan and E. Salari, "Coding of stereoscopic images", SPIE/IS&T Symp. on Electronic imaging, Vol. 1903, Feb. 1993, pp. 228-235.

[93] T.Fujii and H.Harashima, "Data compression of an autostereoscopic 3-D Image", IST/SPIE Symp. on Electronic Imaging, Stereoscopic Displays and applications, Vol.2177, 1994, pp. 108-118.

[94] B. L. Tseng and D.Anastassiou, 'A theoretical study on accurate reconstruction of multiview images based on the Viterbi algorithm', Proc. of Intl. Conf. on Image Processing, Washington D.C, 1995, Vol. 2, pp. 378-381.

[95] H. Aydinoglu and M. H. Hayes, 'Compression of multi-view images', Proc. of Intl. Conf. on Image Processing, Austin, TX, 1994, Vol. 2, pp. 385-388.

[96] A. Puri, R. V. Kollarits and B. G. Haskell, "Stereoscopic video compression using temporal scalability", in SPIE Conf. Visual Commun. Image Processing, 1995, Vol. 2501, pp. 745-756.

[97] A. Puri *et al*., "Compression of stereoscopic video using MPEG-2", Proc. SPIE Vol. CR60, Standards and common interfaces for video information systems, K. R. Rao (Ed.), pp. 309-334.

[98] S. Malassiotis and M. G. Strintzis, "Joint motion/disparity estimation for stereoscopic image sequences", in SPIE Conf. Visual Commun. Image Processing, 1994, Vol. 2308, pp. 614-625.

[99] D. Tzovaras, N. Grammalidis, and M. G. Strintzis, "Object-based coding of stereoscopic image sequences using joint 3D motion/disparity segmentation", in SPIE Conf. Visual Commun. Image Processing, 1995, Vol. 2501, pp. 1678-1689.

[100] M. Ziegler and S. Panis, "An object-based stereoscopic coder", in Intl. workshop on Stereoscopic and Three Dimensional Imaging, 1995, pp. 40-45.

[101] B. L. Tseng and D. Anastassiou, "Perceptual adaptive quantization of stereoscopic video coding using MPEG-2's temporal scalability structure", in Intl. workshop on Stereoscopic and Three Dimensional Imaging, 1995, pp. 52-57.

[102] M. W. Siegel, *et al*., "Compression of stereoscopic image pairs and streams", in SPIE Conf. Stereoscopic Displays and Virtual Reality Systems, 1994, Vol. 2177, pp. 258-268.

[103] S. Sethuraman, M. W. Siegel, and A. G. Jordan, "Multiresolution based hierarchical disparity estimation for stereoscopic image pair compression", in Symposium on Applications of subbands and wavelets, March 1993.

[104] S. Sethuraman, M. W. Siegel, and A. G. Jordan, "A multiresolution framework for stereoscopic image sequence compression", in Proc. of ICIP, 1994, Vol. 2, pp. 361-365.

[105] S. Sethuraman, M. W. Siegel, and A. G. Jordan, "A multiresolutional region based segmentation scheme for stereoscopic image compression", in SPIE Conf. on Digital Video Compression: Algorithms and Technologies 1995, pp. 265-275.

[106] S. Sethuraman, M. W. Siegel and A. G. Jordan, "Segmentation based coding of stereoscopic image sequences", in SPIE Conf. on Digital Video Compression: Algorithms and Technologies 1996, Vol. 2668, pp. 420-429.

[107] Tetsuo Mitsuhashi, "Subjective image position in stereoscopic TV systems - considerations on comfortable stereoscopic images", in SPIE Conf. Human Vision, Visual Processing and Digital Display V, 1994, Vol. 2179, pp.259-266.

[108] V. Grinberg, G. Podnar, and M. W. Siegel, "Geometry of binocular imaging", in SPIE Conf. Stereoscopic Displays and Virtual Reality systems, 1994, Vol. 2177, pp. 56-65.

[109] P. Strobach, "Image coding based on quadtree-structured recursive least squares approximation", Proc. of ICASSP 1989, vol. 4, 1989, pp. 1961-1964.

[110] S. T. Barnard and M. A. Fischler, "Computational and biological theories of stereo", Proc. of the DARPA Image Understanding workshop, September 1990, pp. 439-448.

[111] E. Feig, "A fast scaled-DCT algorithm", in Proc. SPIE Image processing algorithms and techniques, Vol. 1244, 1990, pp. 2-13.

[112] M. Bierling, "Displacement estimation by hierarchical block matching", Visual Commun. and Image processing, Nov. 1988, Vol. 1001, pp. 942-951.

[113] D. G. Jeong and J. D. Gibson, "Lattice vector quantization for image coding", Proc. of ICASSP-89, May 1989, pp. 1743-1746.

[114] T. Aach, *et al.*, "Combined displacement estimation and segmentation of stereo image pairs based on Gibbs random fields", Proc. of ICASSP, 1990, pp. 2301-2304.

[115] R. S. Jasinschi and J. M. F. Moura, "Content-based video sequence representation", Proc. of Intl. Conf. on Image Proc., Oct. 1995, Vol. 2, pp. 229-232.

[116] S. M. Faris, "Micro-polarizer arrays applied to a new class of stereoscopic imaging", SID 91

digest, pp. 840, 1991.

Technical Reports:

[117] A. Cohen, I. Daubechies and J. C. Feauveau, “Biorthogonal bases of compactly supported wavelets”, AT&T Bell Laboratories, TR no. 11217-900529-07.

[118] H. H. Baker, “Depth from edge and intensity based stereo”, TR, Dept. of Computer Sciece, Stanford University, 1982.

[119] M. Okutomi and T. Kanade, ‘A multiple-baseline stereo’, CMU-CS-90-189, November 1990.

[120] T. Nakahara and T. Kanade, ‘Experiments in multiple-baseline stereo’, CMU-CS-93-102, August 1992.

Others:

[121] DISTIMA sequences, generated and distributed under the RACE-DISTIMA European project, October 1994.

[122] J. S. McVeigh, ‘Efficient compression of arbitrary multi-view signals’, Ph.D Thesis, Dept. of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, June 1996.