CARNEGIE MELLON UNIVERSITY

EFFICIENT COMPRESSION OF ARBITRARY MULTI-VIEW VIDEO SIGNALS

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

for the degree

DOCTOR OF PHILOSOPHY
in
ELECTRICAL AND COMPUTER ENGINEERING

by

Jeffrey Scott McVeigh

Pittsburgh, Pennsylvania
June, 1996

Abstract

Multiple views of a scene, obtained from cameras positioned at distinct viewpoints, can provide a viewer with the benefits of added realism, selective viewing, and improved scene understanding. The importance of these signals is evidenced by the recently proposed Multi-View Profile (MVP) extension to the MPEG-2 video compression standard, and their explicit incorporation into the future MPEG-4 standard. However, multi-view compression implementations typically rely on single-view image sequence model assumptions. We hypothesize (and demonstrate) that impressive system bandwidth reduction can be achieved by utilizing displacement vector field and image intensity models tuned to the special characteristics of multi-view video signals.

This thesis focuses on the predictive coding of non-periodic, i.e., arbitrary, multi-view video signals for the applications of simulated motion parallax and viewer-specified degree of stereoscopy. To facilitate their practical use, we desire algorithms that are applicable to the common waveform-based, hybrid encoder framework, which consists of a frame-based prediction followed by residual encoding.

Three novel techniques are developed, which respectively improve the processes of frame-based prediction, residual encoding, and viewpoint interpolation. These are:

- a simple method to adaptively select the best possible reference frame, based on estimated occlusion percentage with the frame to be encoded;
- a low bit rate residual encoding technique that compensates for pixel intensity non-stationarities along a displacement trajectory and for the practical limitations of the prediction process; and
- an algorithm that correctly handles displacement estimation errors, occlusions and ambiguously-referenced image regions for the interpolation of subjectively-pleasing "virtual" viewpoints from a noisy displacement vector field.

We demonstrate the superiority of each of these algorithms on numerous multi-view video signals through comparisons with conventional techniques, and we analyze their cost/benefit ratio in terms of increases in system complexity and storage, offset by rate-distortion improvements. Finally, we indicate the relative significance of these algorithms, and provide insight into how and when they should be combined into a complete, efficient multi-view encoder/decoder system.

Acknowledgments

I would like to thank my co-advisors, Angel Jordan and Mel Siegel, for their support and guidance throughout the course of this work. I am indebted to them not only for their numerous ideas and suggestions, but also for the freedom they gave me to explore topics truly interesting to me. I would also like to thank Siu-Wai Wu and José Moura for participating on my thesis committee. Their advice and comments undoubtedly improved the quality of this thesis.

I extend special thanks to the members of the Advanced Video Display Systems group: Tom Ault, Victor Grinberg, Alan Guisewite, Priyan Gunatilake, Gregg Podnar, Scott Safier, Sriram Sethuraman, and Huadong Wu. Our technical discussions yielded many fruitful ideas that helped to shape and focus my research interests.

I owe a great deal to my parents, extended family, and friends. Although not directly involved with the technical aspects of this work, the emotional and social outlets that they provided me with are largely responsible for the completion of this work.

Most importantly I wish to thank my wife Melissa for her endless encouragement and patience throughout my graduate studies. The joy and laughter she continues to give me are the source of my motivation, and are responsible for keeping me both physically and emotionally fulfilled.

This work was supported by the Advanced Research Projects Agency under ARPA Grant Number MDA 972-92-J-1010.

Contents

Abstract iii

Acknowledgments iv

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Motivation ... 1
1.1.1 Binocular Imagery ... 2
1.1.2 Motion Parallax ... 2
1.1.3 Scene Analysis ... 3
1.1.4 Multi-view Bit Rate ... 3

1.2 Problem Description ... 4

1.3 Related Work ... 5
1.3.1 Hybrid Coders ... 5
1.3.1.1 Frame-based Prediction ... 6
1.3.1.2 Residual Encoding ... 9
1.3.1.3 Irrelevancy Reduction ... 10
1.3.2 Displacement-compensated View Interpolation ... 11
1.3.3 Content-based Coders ... 12

1.4 Contributions ... 13
1.4.1 Optimal Reference Frame Selection ... 13
1.4.2 Restoration-based Residual Coding ... 14
1.4.3 Interpolation from a Noisy Displacement Vector Field ... 14

1.5 Thesis Outline ... 15

2 Multi-view Video Signals 17

2.1 Extension from Single-view Signals ... 17

2.2 Arbitrary Viewpoints ... 21

2.3 Cost Measures ... 24
2.3.1 Bit Rate ... 25
2.3.2 Distortion ... 25
2.3.3 Complexity ... 26

2.3.4 Storage ... 26

2.4 Multi-view Sequence Descriptions ... 27

2.4.1 Cement Mixer Sequence ... 27

2.4.2 Volleyball Sequence ... 29

2.4.3 Finish Line Sequence ... 30

2.4.4 Skater Sequence ... 31

2.4.5 Toy Train Sequence ... 31

2.4.6 Piano Sequence ... 31

2.4.7 Manege Sequence ... 32

2.4.8 Flower Garden Sequence ... 32

3 Theoretical Basis for New Multi-view Algorithms 41

3.1 Image Model ... 41

3.1.1 Unoccluded Regions ... 42

3.1.2 Occluded Regions ... 43

3.1.3 Compact Representation of Image Model ... 43

3.2 Displacement Model ... 46

3.2.1 One-to-One Mapping of Unoccluded Regions ... 48

3.2.2 Occlusion / Displacement Discontinuity Relationship ... 49

3.2.3 Reliability of Block-based Estimates ... 52

4 Optimal Reference Frame Selection 54

4.1 Prediction Performance ... 54

4.2 Multi-view Scenarios ... 55

4.3 Early Thoughts on Selection Algorithms ... 58

4.3.1 Exhaustive Prediction ... 58

4.3.2 Occlusion-based Selection ... 59

4.4 New Algorithm ... 59

4.4.1 Composite Displacement Vector Fields ... 62

4.4.1.1 Single-step Predictions ... 62

4.4.1.2 Unoccluded Region Detection ... 65

4.4.1.3 Vector Field Addition ... 68

4.4.2 Occlusion Measure ... 69

4.5 Cost Analysis ... 70

4.5.1 Complexity ... 71

4.5.2 Storage ... 73

4.6 Experimental Results ... 74

4.7 Summary ... 84

5 Restoration-based Residual Coding 85

5.1 Residual Coding ... 86

5.2 Predicted - Residual Image Correlation 86

5.3 Conventional Techniques 93

5.3.1 Transform-based with Scalar Quantization 93

5.3.2 Vector Quantization 94

5.4 New Approach: Restoration-based Coding 96

5.4.1 Scalar Restoration (SR) 99

5.4.2 Vector Restoration (VR) 103

5.4.3 Interpretation and Comparison 104

5.4.4 Extensions 107

5.5 Cost Analysis 108

5.5.1 Complexity 108

5.5.2 Storage 110

5.6 Experimental Results 112

5.7 Summary 122

6 Interpolation from a Noisy Displacement Vector Field 124

6.1 Viewpoint Interpolation 125

6.1.1 Motivation 125

6.1.2 Unoccluded Regions 127

6.1.3 Effects of Displacement Estimation Errors and Occlusions 129

6.2 Prior Work 131

6.3 Improved Interpolation 134

6.3.1 Field Refinement via Self-Synthesis 137

6.3.2 Displacement Selection for Ambiguous Regions 142

6.3.3 Inferring Range Information for Occlusions 145

6.4 Cost Analysis 148

6.4.1 Complexity 148

6.4.2 Storage 152

6.5 Experimental Results 153

6.6 Summary 157

7 Conclusions and Future Work 180

Bibliography 184

List of Tables

Table 2.1: Sequence highlights 28

Table 4.1: Optimal reference frames for View 0 in simulated multi-view signal depicted in Fig. 4.1. 58

Table 4.2: *Flower Garden* sequence's original temporal indices for simulated two-view signal. 75

Table 4.3: Adaptively selected reference frames for the original frames in View 1 of the simulated signal. 75

Table 4.4: Reference frame ranking, with respect to prediction PSNR, for the *Skater* sequence 80

Table 4.5: Average prediction PSNR and coded bit rate for fixed, exhaustive search, and adaptively-selected reference frame schemes. 82

Table 4.6: Number of times a relative reference frame location was selected for the prediction of the two dependent views, $F(1,n)$ and $F(2,n)$, a) exhaustive search technique, b) occlusion-based, adaptive technique. 83

Table 5.1: Per-pixel complexity comparison for coding schemes. 110

Table 5.2: Storage requirement comparison for coding schemes (in megabytes). 112

Table 6.1: Average fixed complexity costs for the generation of interpolated displacement vector field. 149

Table 7.1: Relative encoder cost comparison summary between fixed, exhaustive, and adaptive reference frame selection schemes. 182

Table 7.2: Relative encoder cost comparison summary between vector quantization and vector restoration residual encoder implementations. 182

Table 7.3: Relative decoder cost comparison summary between basic and enhanced viewpoint interpolation techniques. 182

List of Figures

Figure 1.1: Camera configuration for two-dimensional motion parallax application. The cameras (C_{ij}) are arranged with parallel axes and coplanar image sensors........... 3

Figure 1.2: Frame-based hybrid encoder and decoder structures. 7

Figure 2.1: Camera viewpoint and relationship between world-coordinates and pixel location. 19

Figure 2.2: Periodic, horizontally-spaced views, a) twenty stacked images, b) epipolar plane image (EPI) for middle row of stacked images. 23

Figure 2.3: Camera Parameter Set A. 29

Figure 2.4: Camera Parameter Set B. 30

Figure 2.5: *Cement Mixer* sequence, frame 50. 33

Figure 2.6: *Volleyball* sequence, frame 200. 34

Figure 2.7: *Finish Line* sequence, frame 100. 35

Figure 2.8: *Skater* sequence, frame 175. 36

Figure 2.9: *Toy Train* sequence, frame 150. 37

Figure 2.10: *Piano* sequence, frame 50. 38

Figure 2.11: *Manege* sequence, frame 25. 39

Figure 2.12: *Flower Garden* sequence, frames 75 and 77. 40

Figure 3.1: $\aleph_4$ (a) and $\aleph_8$ (b) two-dimensional, noncausal neighborhoods. 44

Figure 3.2: Relationship between occlusion and unoccluded region displacement discontinuity, a) sample frames with constant displacement for background and foreground object, b) vertical component of actual displacement along middle image column. 51

Figure 4.1: Two-view scenario illustrating time-varying location of the optimal reference frame. 57

Figure 4.2: Flow-chart for adaptive reference frame selection algorithm. 61

Figure 4.3: Three-view reference frame location scenario, a) single-step predictions, b) possible reference frames for a maximum temporal-offset of two frames. 63

Figure 4.4: Displacement vector thresholding and field reversal, a) original field, b) field after SNR thresholding, c) reversed field. 66

Figure 4.5: Generation of composite displacement vector field through field addition, a) and b) single-step displacement fields, c) composite field. 68

Figure 4.6: Temporal location of possible references for $F(1,50.5)$ in *Finish Line* sequence. 76

Figure 4.7: Horizontal (left-column) and vertical (right-column) displacement components for *Finish Line* $F(1,50.5)$, a) single-step field from $F(0,51)$, b) processed single-step field, c) composite field for $F(0,50)$, d) composite field for $F(1,49.5)$. 77

Figure 4.8: Graphical display of estimated displacement vectors for $F(1,50.5)$, a) reference $F(0,51)$, b) reference $F(1,49.5)$. 78

Figure 4.9: *Finish Line* performance for possible reference frames, a) prediction PSNR, b) total coded bits per frame. 79

Figure 4.10: *Skater* prediction PSNR comparison of fixed vs. adaptive reference frame selection for, a) entire sequence, b) last 100 frames. 81

Figure 5.1: Illustration of predicted-residual image correlation for *Manege* sequence, a) original $F(0,45)$, b) decoded reference $F'(1,45)$, c) predicted image, d) absolute error image. 91

Figure 5.2: Frame-based hybrid encoder and decoder structures with "prediction-directed" residual coding. 97

Figure 5.3: Typical pixel neighborhoods in predicted image for the restoration of pixel location (u, v) in the residual image. Each neighborhood is denoted by the union of shaded circles. 100

Figure 5.4: Vector restoration encoder and decoder structures. 106

Figure 5.5: PSNR vs. average rate for *Manege* training set, a) balanced, binary trees, b) length-pruned trees. 115

Figure 5.6: Per-frame performance for View 0 of *Manege* sequence, a) PSNR, b) bits used to encode the luminance image component. 116

Figure 5.7: Rate-distortion comparison for View 0 of *Piano* sequence, a) PSNR, b) total coded bits per frame. 118

Figure 5.8: Rate-distortion comparison for View 1 of *Piano* sequence, a) PSNR, b) total coded bits per frame. 119

Figure 5.9: Rate-distortion comparison for View 0 of *Finish Line* sequence, a) PSNR, b) total coded bits per frame. 120

Figure 5.10: Rate-distortion comparison for View 1 of *Finish Line* sequence, a) PSNR, b) total coded bits per frame. 121

Figure 6.1: Multi-view system for simulated motion parallax and viewer-specified inter-camera separation, (a) functional block diagram, (b) virtual camera positions of desired intermediate views. 126

Figure 6.2: Interpolated displacement vector field along epipolar line. 129

Figure 6.3: (a) Actual displacement field, (b) estimated field with occluded (blackened) and ambiguously-referenced (lightly-shaded) image regions. 131

Figure 6.4: Flow-chart for improved viewpoint interpolation technique. 138

Figure 6.5: Self-synthesis operation along single epipolar-line, (a) actual displacement field, (b)-(d) synthesis of View 0 from View 1, (e)-(g) synthesis of View 1 from View 0, (h) final processed field. 143

Figure 6.6: Inferring range information for occluded regions, (a) perspective project of sample scene, (b) field interpolated from unoccluded regions, (c) interpolated field after filling occluded regions. 147

Figure 6.7: *Flower Garden*, (a) original frame 0, (b) original frame 3 159

Figure 6.8: Estimated displacement fields (horizontal component) after various processing stages. Left column: frame 0 from frame 3. Right column: frame 3 from frame 0. (a) after thresholding, border elimination and field reversal, (b) after self-synthesis of frame 0, (c) after self-synthesis of frame 3 160

Figure 6.9: Occlusion localization results, (a) frame 0, (b) frame 3. 161

Figure 6.10: *Flower Garden* frame 1, (a) original, (b) interpolated. 162

Figure 6.11: Absolute frame difference for frame 1, (a) frame repetition, PSNR=15.72 dB, (b) interpolation, PSNR=28.14 dB. 163

Figure 6.12: *Flower Garden* frame 1, (a) interpolated without performing displacement field processing and ambiguous region selection (PSNR=24.54 dB), (b) bidirectionally-predicted from reference frames 0 and 3 using a fixed-size, block-based technique (PSNR= 28.55 dB). 164

Figure 6.13: *Flower Garden* frame 2, (a) original, (b) interpolated. 165

Figure 6.14: Absolute frame difference for frame 2, (a) frame repetition, PSNR=15.85 dB, (b) interpolation, PSNR=28.00 dB. 166

Figure 6.15: *Flower Garden* frame 2, (a) interpolated without performing displacement field processing and ambiguous region selection (PSNR=24.86 dB), (b) bidirectionally-predicted from reference frames 0 and 3 using a fixed-size, block-based technique (PSNR= 28.89 dB). 167

Figure 6.16: *Volleyball*, (a) original $F(0,155)$, (b) original $F(1,155)$ 168

Figure 6.17: *Volleyball* $F(0.5,155)$, (a) interpolated using enhancements, (b) interpolated without enhancements. 169

Figure 6.18: *Cement Mixer*, (a) original F(1,175), (b) original F(2,175) 170

Figure 6.19: *Cement Mixer* $F(1.25,175)$ interpolated using improved technique 171

Figure 6.20: *Cement Mixer* $F(1.5,175)$ interpolated using improved technique 172

Figure 6.21: *Cement Mixer* $F(1.75,175)$ interpolated using improved technique 173

Figure 6.22: *Cement Mixer* $F(1.5,175)$ interpolated without displacement field processing.. 174

Figure 6.23: *Cement Mixer* $F(1.5,175)$ interpolated using range ordering for ambiguously-referenced image regions and zero displacement occlusion filling. 175

Figure 6.24: *Cement Mixer* $F(1.5,175)$ interpolated using unprocessed field, range ordering for the selection of ambiguously-referenced image regions, and filling of holes with zero displacement. 176

Figure 6.25: *Piano*, (a) original $F(0,25)$, (b) original $F(1,25)$ 177

Figure 6.26: *Piano* $F(0.5,25)$, (a) interpolated using improved technique, (b) interpolated using basic scheme. 178

Figure 6.27: Close-ups of *Piano* frame, (a) original $F(0,25)$, (b) interpolated $F(0.5,25)$ using improved technique (c) interpolated $F(0.5,25)$ using basic interpolation scheme, (d) original $F(1,25)$ 179

Figure 7.1: Complete multi-view hybrid coder structure incorporating adaptive reference frame selection, restoration-based residual coding, and viewpoint interpolation from a noisy displacement vector field. 183

Chapter 1

Introduction

We present a set of novel techniques for the efficient, predictive coding of video signals obtained from multiple cameras positioned at arbitrary viewpoints. While multi-view video signals can provide the benefits of added realism, selective viewing, and improved scene understanding, an extremely large number of bits is required to describe these multidimensional signals in their raw, uncoded form. The inherent redundancy within these signals must be fully exploited to allow for their practical use.

The three algorithms developed in this thesis address the special characteristics of multi-view signals and respectively improve the processes of prediction, residual encoding, and image interpolation within a hybrid encoder framework by: 1) adaptively selecting the best possible reference frame, 2) exploiting the predicted-residual image correlation due to practical limitations of the prediction operation, and 3) accurately interpolating occluded and ambiguously-referenced image regions from a noisy displacement vector field. We evaluate the superiority of these algorithms on numerous multi-view video signals through comparisons with conventional encoding techniques, and we analyze their cost/benefit ratio in terms of increases in system complexity and storage, offset by rate-distortion improvements. These techniques may be performed independently of one another or they may be concatenated into a complete multi-view system. We provide insight into how the bit rate constraint and application for a particular signal dictate the use of one or more of these algorithms.

1.1 Motivation

A video signal provides an enormous amount of information on the visual content of the real-world. This information depends on both the contents of the scene and also the parameters of the camera that generated the signal. However, a video signal obtained from a single camera can provide visual information about the scene from only one particular viewpoint/scale/spectral-band at any given time instant – certain visual communication applications require more information. Multiple views of the scene can provide the benefits of added realism, selective viewing, and improved scene analysis. The motivation for the efficient compression of multi-view signals next will be provided through the development of example applications that illustrate these benefits.

1.1.1 Binocular Imagery

An obvious application requiring more than one view of the scene is that of a binocular imaging system. Binocular imagery, which, on a suitable display, precisely replicates the geometry of human vision, is obtained from two identical cameras separated by a horizontal distance, which is equal to the inter-ocular separation (typically 65 mm), and oriented with parallel camera axes and coplanar image sensors [33]. By presenting the appropriate views to the corresponding left- and right-eyes of a viewer, two slightly different perspective views of the scene are imaged on each retina. The brain then fuses these images into one view and the viewer experiences the sensation of stereopsis, which provides added realism through improved depth perception [45, 95].

1.1.2 Motion Parallax

For further improvements in realism, the binocular imaging system can be extended to provide the viewer with the depth cue of motion parallax [74, 99].[1] Motion parallax, which provides the distinction between binocular and three-dimensional imagery, can be simulated by obtaining multiple, closely-spaced views of the scene and then presenting the appropriate binocular image pair based on the viewer's position. Psychovisual studies of the capabilities of the human visual system to discern variations in object displacement indicate that over ten distinct views per inter-ocular separation are needed to ensure smooth and realistic motion parallax [51, 75].[2] The camera configuration of a system capable of providing both horizontal and vertical motion parallax is shown in Fig. 1.1. In this scenario, $I \times J$ cameras are positioned on a two-dimensional grid, with camera spacing of Δc_x in the horizontal direction and Δc_y in the vertical direction.

---

1. Motion parallax is the phenomenon whereby a change in the viewer's position results in objects appearing to move, with the amount of displacement inversely related to the object's distance from the viewer.
2. Specifically, the required number of views depends on the depth range of the scene and the visibility threshold for parallax shifts. A threshold of 1.15 min of arc was estimated experimentally in [75].

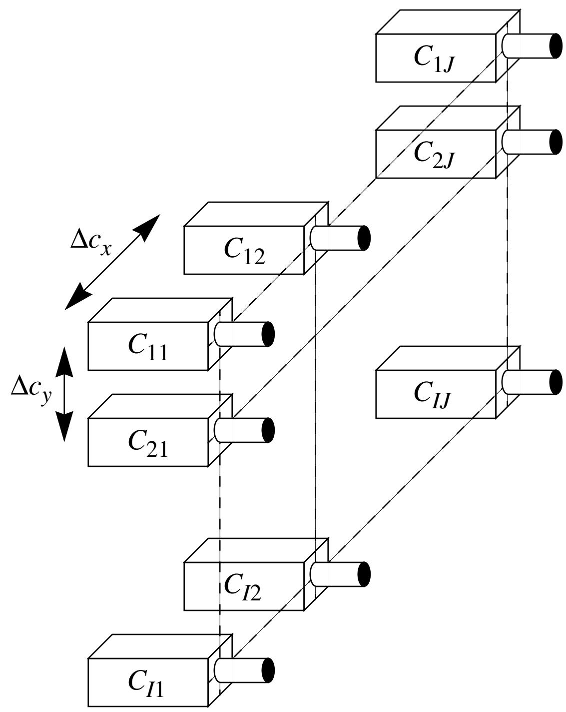


Figure 1.1: Camera configuration for two-dimensional motion parallax application. The cameras (C_{ij}) are arranged with parallel axes and coplanar image sensors.

1.1.3 Scene Analysis

Multiple views of the scene are also useful in applications that analyze the three-dimensional structure of the scene. Typical applications include vision-based navigation systems [34, 52, 91], scene modeling [46], and object-based compression [61, 72, 94]. If the configuration of the cameras capturing the multiple views are known, a straightforward relationship can be used to estimate the range of objects from the displacement of corresponding points within the views [92]. Only two views of the scene are required to calculate displacement information for unoccluded regions. Additional views are useful for autocalibration [20, 107], reducing ambiguity in the displacement estimation process [37, 89], and minimizing the number and area of occluded regions [30].

1.1.4 Multi-view Bit Rate

The described applications are but a sampling of the possible uses for multi-view video signals. Additional applications may use cameras that vary not only in their viewpoint, but also have varying scale [101] and spectral bandwidth selectivity parameters [5, 35]. Regardless of their ultimate use, it is likely that these signals will need to be transmitted and/or stored in order to be displayed at a remote location, re-displayed at a future time, or processed off-line.

In its raw form, the bit rate for a multi-view signal is given by the simple equation,

$$\text{bit rate} = (\text{\# of views}) \times (\text{frames per second}) \times (\text{pixels per frame}) \times (\text{bits per pixel}) \tag{1.1}$$

For common values of 30 frames per second (fps), 640×480 pixels per frame (ppf), and 24 bits per pixel (bpp), the uncoded bit rate for a signal with three views is over 6.6×10^8 bits per second (bps). Using the digital video transmitter described in [38], a compression factor of at least 33:1 is required to broadcast this multi-view signal in a 6 MHz channel, with a service area comparable to the NTSC service area.

Since visual communication systems are often constrained by their transmission bandwidth and storage requirements, it is imperative that the bit rate for multi-view signals be reduced through the removal of redundant and irrelevant information within and between the various views.

1.2 Problem Description

We will examine the problem of efficiently compressing multi-view video signals, where the cameras are restricted to vary only in their viewpoint; i.e., all cameras have identical scale and spectral bandwidth parameters. As a point of reference, the example applications described in Section 1.1 satisfy this constraint.

We will strive for near-term solutions, in the sense that the compression algorithms developed or enhanced should be realizable with current, available technology suitable for mass-market production. This goal not only attempts to limit the complexity and related cost of systems that utilize these algorithms, but also ensures that any additional costs will be commensurate with the benefits provided by multi-view signals. Therefore, we will remain within a *hybrid encoder* framework (see Section 1.3.1), and we will develop techniques that are consistent with this methodology and exploit the special characteristics of these signals.

We will measure the success of the algorithms developed in this thesis through the improvement in rate-distortion performance over both conventional, single-view techniques applied to each view and techniques that explicitly address multi-view signals. We will also conduct resource requirement analyses to satisfy our realizability constraint. System designers should find this cost/benefit analysis valuable when they decide whether to implement a particular algorithm.

1.3 Related Work

This section describes work related to the problem of efficiently compressing multi-view video signals. Often, this work has amounted to merely replicating single-view systems. The merits of the various techniques are presented, and areas that warrant additional research are highlighted. In particular, since we are dealing with the hybrid coder framework, the methodology and limitations of this approach will be discussed in detail.

1.3.1 Hybrid Coders

A common method for the compression of both single- and multi-view video signals is that of a frame-based hybrid coder. The hybrid nature of these coders is their combination of both frame-based prediction and residual encoding, which respectively exploit redundancy between (inter) and within (intra) frames. Interframe redundancy is due to the predictable displacement of objects between frames, while intraframe redundancy is due to the spatial correlation of pixels. Popular single-view implementations of hybrid coders are found in the MPEG-1, MPEG-2, H.261, and H.263 video compression standards [1, 2, 3, 4]. Frame-based hybrid encoders view a video signal as a set of image frames. The signal is compressed by performing intraframe coding on marker frames, and by encoding all other frames using information from previously decoded frames. The marker frames are used for initialization, random-access and re-synchronization purposes.

Additional views are easily incorporated into this methodology due to the frame-based approach, where the number of frames increases linearly with the number of views. This extension of the hybrid coder structure has been reported in numerous systems dealing with multi-view signals [15, 59, 78, 86, 87]. The frames from each view may be treated equally or an implicit ranking may be established between the individual views. The latter approach is taken by the recently proposed multi-view extension to the MPEG-2 standard, which uses the temporal-scalability option of the standard to accommodate multiple views of the scene [77]. The MPEG-2 Multi-View Profile (MVP) defines one view as the base layer and all additional views as enhancement layers. This ranking is used to ensure compliance with the MPEG system-layer structure and to avoid any restriction on the number of allowable views.

Frame-based hybrid encoder and decoder structures are depicted in Fig. 1.2, where the prediction and residual encoding/decoding functional blocks may contain numerous lower-level operations. While these processes are highly interrelated, most multi-view techniques known to us treat them independently. We next describe work performed on the compression of multi-view signals with respect to the prediction and residual encoding operations, while remaining cognizant of their interdependence. We follow this by examining techniques for removing irrelevant information from these signals.

1.3.1.1 Frame-based Prediction

The prediction process reduces interframe redundancy by requiring only new information between frames to be encoded. The prediction operation typically amounts to the specification of a displacement model and the tracking of corresponding regions between frames.[3] The degree of interframe redundancy in a single-view signal is related to the temporal sampling rate, the motion of the camera, and the motion of objects within the scene. For relatively high sampling rates and moderate camera and object motion, the compression gain achievable from the prediction process is quite impressive (an order of magnitude increase in the prediction peak signal-to-noise ratio over simple frame repetition is common [31, 44]). Similar gains have been reported for multi-view signals, where the frame-based prediction is generated from a reference frame offset in time and/or viewpoint [5, 15, 23, 59, 86, 102]. For these systems, interframe redundancy is related not only to the temporal sampling rate and object motion but also to the scene structure and camera configuration.

For multi-view signals, the number of bits needed to describe the prediction can be reduced if the reference frame is offset from the desired frame only in viewpoint. Since the prediction is described by the displacement of regions between the reference and desired frame, fewer bits are needed if the displacement range is restricted. The displacement range of corresponding points between two views obtained at the same time instant is constrained to one-dimension, which is referred to as the epipolar line [9, 11, 56, 92].[4] Since the camera configuration is rarely known to

---

3. Due to their overwhelming use, we will consider only block-based (fixed and variable sized) displacement compensated prediction schemes in this thesis. As such, we shall interchangeably use the terms prediction process and displacement estimation/compensation.

4. Assuming a pinhole camera model, a point in three-dimensional space will project its image at the intersection of the camera's sensor plane with the line passing through the point and the camera lens. The relative location of the corresponding projection in another view must lie along the epipolar line, which is the intersection of the other camera's sensor plane with the plane containing the two camera lenses and the image point.

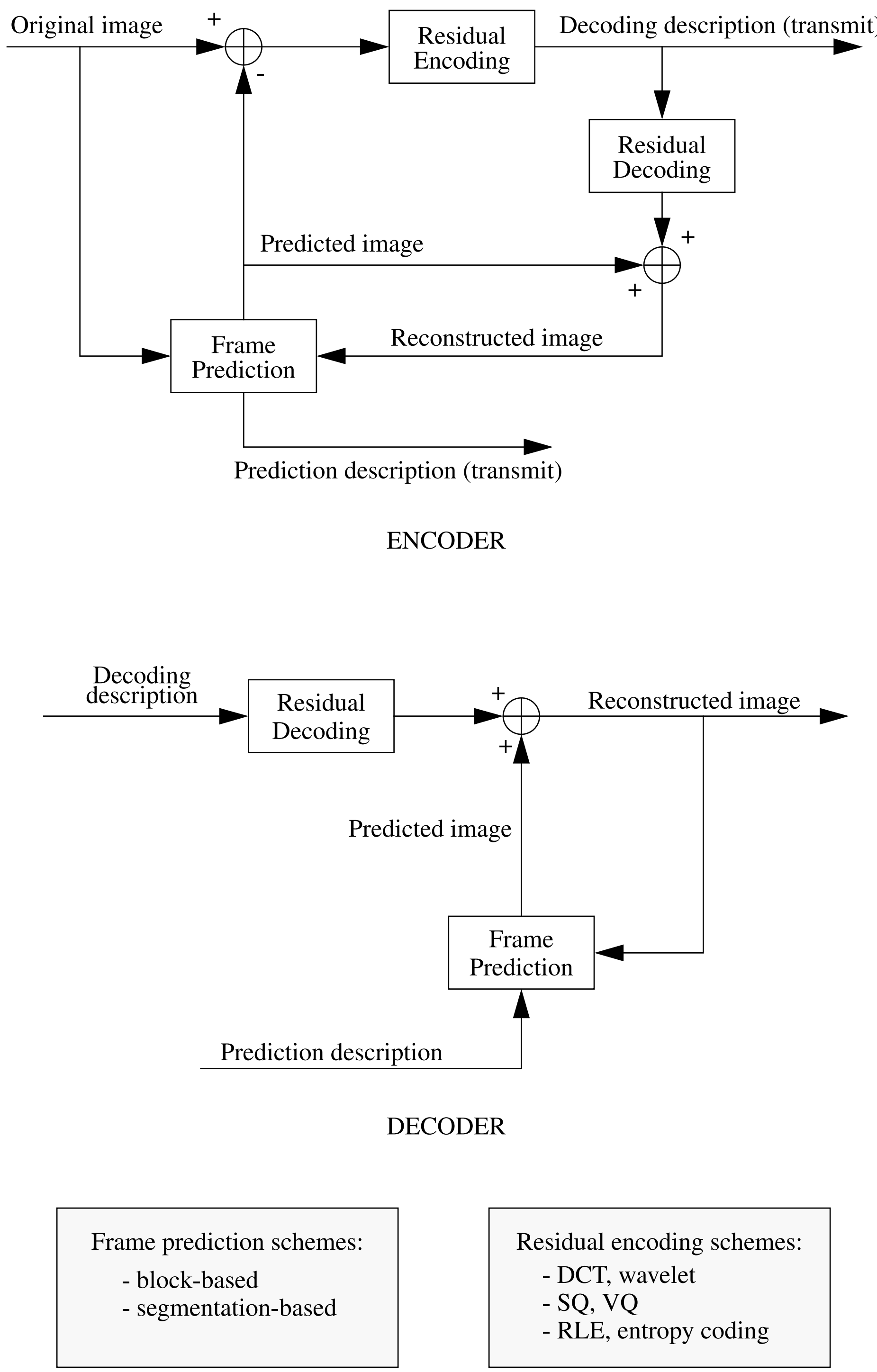


Figure 1.2: Frame-based hybrid encoder and decoder structures.

the level of accuracy required to exactly specify the epipolar line, most multi-view prediction schemes allow for some deviation in the displacement range from the 1-D constraint.

A substantial body of work has been devoted to improving the prediction performance for multi-view signals through segmentation-based schemes, which are applicable to both single- and multi-view signals [14, 87, 100]. These techniques begin to approach the concept of content-based coding (see Section 1.3.3) by predicting homogeneous regions within each frame from a reference frame, as opposed to purely pixel-based schemes, such as the ubiquitous block-based techniques. Regions are segmented and classified based on numerous attributes, including their displacement between frames. The standard approach for these techniques is a split-and-merge process whereby larger-sized regions are split using a multiresolutional decomposition, and regions with similar attributes are merged at the higher levels of resolution. The attributes may be tracked through time to avoid the computational cost of performing the segmentation and to reduce the number of bits needed to describe the region structures.

Superior frame predictions are possible due to the signal dependent nature of the segmentation-based algorithms: areas uniform in the selected attributes are segmented into larger-sized regions, requiring fewer bits per pixel to describe their displacements; more bits are allotted to detailed areas through their segmentation into smaller-sized regions. The improved prediction has been shown to significantly reduce the number of bits needed in the residual encoding stage [14, 87, 100]. The more accurate displacement estimation afforded by these techniques also can be used to improve displacement-compensated interpolation schemes (see Section 1.3.2). However, segmentation-based methods are not without their faults. If the scene contains numerous, high-detail regions, the coder's bit rate actually may increase over pixel-based prediction schemes. Also, the split-and-merge process is often quite heuristic and may require fine tuning to yield an adequate segmentation. Nevertheless, segmentation-based prediction likely will be an integral portion of future multi-view encoding systems.

One seemingly glaring omission in both the displacement-range-constraint and the segmentation-based-prediction bodies of work is that the reference frames used to predict the desired frame are often heuristically chosen and pre-selected. The performance of frame-based predictors is related to the notion of "similarity" between the desired and reference frames. For single-view signals, the most similar reference frame is generally the temporally nearest decoded frame. However, the relative location of the optimal reference frame for multi-view signals is not as straightforward; it depends on the structure and motion of both scene objects and cameras, and it varies as the signal evolves and progresses. The gain that would be achieved through the use of the most similar reference frame is independent of the type of frame-based prediction used [64]. Puri, *et. al*, allude to the effect of reference frame configuration on coding performance for stereoscopic video in [78]; yet, they do not discuss how the optimal reference should be selected. The development of an algorithm to overcome this shortcoming is one of the major contributions of this thesis.

1.3.1.2 Residual Encoding

Ideally, the difference (residual) between the original and predicted images would be spatially-uncorrelated. In fact, the underlying content of the original image can always be seen in the residual image, and traditional still-image compression techniques then can be applied to exploit this correlation. The residual encoder may include numerous operations, such as: the transformation of the residual image data to a generalized frequency domain (DCT [18] and wavelet transformations [49]), and the application of both lossy (scalar and vector quantization [32]) and lossless (run-length and entropy coding [42]) compression techniques. The choice of which technique(s) to use depends on the bit rate, complexity and reconstruction fidelity constraints of the system along with the signal content.

In most hybrid coders, the majority of the bits needed to describe the coded signal are consumed by the residual encoding process (for a high fidelity coded sequence, the residual coding may require over 90% of the total bit rate). The exact number of bits required to reconstruct the original image to a desired level of quality depends on the accuracy of the prediction and the rate-distortion characteristics of the residual encoder. Due to the feedback nature of the hybrid coder, this relationship also holds in the opposite direction; i.e., prediction performance is related to the quality of the reconstructed image.

The majority of work dealing with the residual encoding process for multi-view signals has amounted to: 1) generating a superior prediction to limit the number of bits needed to describe the residual image (see Section 1.3.1.1), and/or 2) performing more severe, lossy compression on individual views that are considered less subjectively-important than other views (see Section 1.3.1.3). Even with improvements in the prediction process, it is unlikely that a system capable of yielding "perfect" frame predictions, in the sense that the residual image contains only irrelevant information, will be obtainable. Therefore, we feel that further work on improving the residual encoding process is needed to adequately reduce the extremely large bit rates of multi-view video signals.

A possible area for improvement becomes apparent when the residual encoder model is examined in detail. One universal characteristic of all residual encoding schemes known to the author is that the residual image is operated upon independently of any knowledge of the predicted image.

This approach is based on the model that the original image consists of both a predictable portion and an unpredictable, white-noise component [26]. The prediction operation is assumed to exactly yield the predictable information of the original image and the residual image merely contains the white-noise process. Therefore, the residual and predicted images are uncorrelated and no coding gain is possible through the use of information available in the predicted image. However, due to the bit rate and complexity constraints imposed on the prediction process and the non-stationarity of the image data, this model becomes invalid: substantial correlation exists between the predicted and residual images. This is particularly true when the coder operates in low bit rate regions [29].

A substantial contribution of this thesis is the development of a novel coder model that exploits this correlation, due to the practical constraints on the prediction process, to achieve superior rate-distortion performance over traditional residual encoders. A technique similar to our solution was reported independently by Richardson in [84]; however, this work was concerned with the problem of restoring a degraded image using only a model of the degradation process and did not address the coding problem.

1.3.1.3 Irrelevancy Reduction

Irrelevancy reduction is the final form of information reduction in hybrid coders that will be examined. The classification of information as being irrelevant depends on the use of the information. When video is to be viewed by a human observer, characteristics of the human visual system (HVS) can be exploited to remove information that is imperceptible to the viewer. For single-view systems, irrelevancy reduction typically amounts to performing spatial- and temporal-frequency dependent quantization. Example techniques include devoting more bits to lower spatial-frequency components of the signal, subsampling the chrominance information, and relaxing fidelity requirements for high-motion regions [60].

These methods may be applied to multi-view signals intended for viewing by a human observer. Also, binocular systems often exploit the singularity-of-vision property of the HVS to remove irrelevant information [22]. In binocular vision, one eye is dominant over the other eye – artifacts viewed by the non-dominant eye will be masked by the dominant eye. This feature allows for a non-symmetric allocation of bits between the two views; i.e., one view is compressed much more coarsely than the other view. Problems with this approach arise since not all viewers have the same dominant eye. Also, it is unclear how more than two views would be ranked, and whether a classification and subsequent differential compression of multiple views is justifiable. Therefore, we wish to develop multi-view algorithms that do not rely on the singularity-of-vision property to achieve an improved compression ratio.

1.3.2 Displacement-compensated View Interpolation

A related technique for the compression of multi-view video signals is that of intermediate view synthesis or interpolation. Here, a minimal set of views that represent the structure of the scene are coded using a conventional hybrid encoder. Additional views, interior to the given views, are then interpolated at the decoder by mapping pixel intensities based on the displacement of image regions. Viewpoint interpolation is important in numerous applications, such as the system capable of simulating horizontal and vertical motion parallax, which would require an excessive camera complexity and bit rate if all views were captured, encoded and transmitted. It would be much more efficient to capture and transmit only the views on the four corners of the two-dimensional grid, and then to synthesize the other views at the decoder.

Numerous techniques for the synthesis of intermediate views have been reported in the literature. A common approach is through the generation of epipolar plane images (EPI) [30, 40, 47]. The EPI-based techniques use multiple, closely-spaced views of the scene to minimize the occurrence of occlusion and reduce errors in the displacement estimation process; however, if an accurate model of the displacement of objects between views is known, the generation of multiple views amounts to oversampling the scene and is inherently redundant. The other class of techniques for the synthesis of intermediate views is that of displacement-compensated view interpolation, which is closely related to motion-compensated interpolation of single-view signals for frame rate conversion, de-interlacing and frame skipping [13, 41, 47, 48, 83, 93]. Here, the intermediate view is synthesized by estimating the displacement of regions between two views and linearly scaling this displacement based on the relative viewpoint of the desired view between two given views. This operation is straightforward and accurate for unoccluded regions where the exact displacement is known. The performance of displacement-compensated interpolation schemes degrades abruptly when a region is visible in only one view (occluded) and when displacement estimation errors occur [63]. Techniques for handling occlusions in single-view signals are provided in [83] and [93]. However, these methods are not applicable to multi-view imagery, and they do not achieve an adequate delineation of occlusion borders to avoid a subsequent residual encoding stage for image pairs with relatively large displacement ranges.

Since we wish to eliminate the generation and transmission of the interior views for the compression of multi-view signals, we do not have the luxury of performing residual encoding to reconstruct the desired view if the interpolation process yields an unsatisfactory image. We also wish to synthesize intermediate viewpoints using only the information available from the decoding of a representative set of exterior views. We address the problem of improving the interpolation performance for occluded and ambiguously-referenced regions from a noisy displacement vector field by using knowledge of the camera configuration and the displacement estimation process. The development of this enhancement, which provides improved interpolation for these "difficult-to-synthesize" regions, is the final contribution of this thesis.

1.3.3 Content-based Coders

A recent trend in video compression algorithms is the shift from frame-based methodologies to techniques that treat video as a combination of visual objects. In its simplest form, content, or object, based compression entails the segmentation of image frames into "objects" that have some common attributes or features, which may or may not correspond to physical objects (see Section 1.3.1.1). More sophisticated content-based coders attempt to decompose the video signal into real-world objects. Once the signal has been decomposed, the object models are sent to the decoder and the video signal is reconstructed from a script describing how the objects evolve as the signal progresses. Content-based coding is at the core of the MPEG-4 video compression standard currently under development, which contains a provision for the compression of multiple, concurrent views of the scene [81].

This object-based methodology efficiently handles multi-view signals since an object visible in separate views can be represented compactly as a single object, and the various perspective views can by synthesized from the object model and knowledge of the camera viewpoints. This capability also can be used to generate views from "virtual" viewpoints, perform content-based retrieval, and actually alter the content and evolution of the video signal [81].

Unfortunately, the availability of systems employing content-based compression is limited due to problems with robustness and complexity; user-intervention is often required to obtain subjectively-acceptable coded video and the operations of object detection, segmentation, and tracking are time-consuming and do not easily facilitate parallel implementations, as do the rote frame-based techniques. Also, object-based systems do not scale well to multiple objects.

We take the stance that, while content-based coders may eventually provide a very efficient representation of multi-view video signals, real-time and robust systems are still quite a few years away from reality. To improve upon current multi-view visual communication systems employing hybrid coders and also future content-based coders, we will address the problems of: 1) selecting the best possible reference frame for the prediction/segmentation process, 2) fully exploiting the residual-predicted image correlation to improve the performance of the residual encoder, responsible for handling situations when the prediction/segmentation process is imperfect, and 3) interpolating virtual viewpoints based on noisy displacement vector fields obtained from frame-based displacement estimation techniques.

1.4 Contributions

This section details the contributions of this thesis for the concurrent compression of multiple, interrelated video signals. For clarity, the contributions are separated into the scope of the specific algorithms presented. Individually, these algorithms are valuable video compression tools; together, they represent a complete and efficient methodology for the compression of multi-view signals.[5]

1.4.1 Optimal Reference Frame Selection

We present a new algorithm that adaptively selects the best possible reference frame for the predictive coding of multi-view video signals, based on estimated occlusion percentage with the desired frame. We illustrate the effect of occlusion on prediction performance, and develop the underlying principle that relates occlusion and the discontinuity between displacement vectors of adjacent, unoccluded image regions. We provide a method to estimate the relative amount of occlusion between two frames from the variance of a composite displacement vector field. Single-step displacement fields are combined to generate the composite fields, which eliminates the computationally costly process of displacement estimation for each candidate reference frame. The reference frame with minimum occlusion (maximum similarity) then is used in the prediction of the desired frame.

The performance increase obtainable from using this algorithm is obviously signal-dependent. When compared with non-adaptive techniques, the adaptive algorithm achieved up to 4 dB improvement in frame prediction performance and required 10-30% fewer bits for the residual encoding process, for the set of multi-view signals examined in this thesis. These gains are obtained with acceptable complexity and storage costs.

5. These algorithms have been implemented in a software simulation capable of performing any combination of the described techniques, along with other conventional schemes. All experimental results were obtained from this simulation.

1.4.2 Restoration-based Residual Coding

We develop a novel class of residual coders that adaptively exploits the non-trivial correlation between the predicted and residual images. This concept represents a significant addition to the theory of predictive coding. We present a stochastic model of the multi-view video signal and illustrate how the practical constraints imposed on the prediction process and the non-stationarity of the signal preclude the residual image from being uncorrelated with the predicted image. The *vector restoration* coder involves the application of optimal operations to predicted image blocks (vectors) with the goal of restoring the original image block. Compression is achieved by only transmitting the index of the restoration operation from a codebook of operations for each predicted image block.

Vector quantization is shown to be a special, and generally sub-optimal, case of vector restoration. In fact, in the unlikely event that the predicted and residual images are uncorrelated, these techniques are equivalent. By using the vector restoration coder, the bit rates for the signals examined were reduced by 15% when compared to a vector quantization implementation. Similar, although not as impressive, improvements were observed over DCT-based residual coders. However, the restoration-based encoder requires a substantial increase in storage and complexity costs over the VQ and DCT approaches, and is thus most suitable for applications with severe bit-rate constraints.

1.4.3 Interpolation from a Noisy Displacement Vector Field

We enhance displacement-compensated interpolation algorithms by accurately handling displacement estimations errors, occlusions, and ambiguously-referenced regions. We examine the displacement estimation process and illustrate the causes of estimation errors. We detect occluded regions and displacement estimation errors through a displacement estimation confidence measure. The location of displacement discontinuities, or equivalently occlusions, are finely delineated through the process of *self-synthesis*, which checks the accuracy of the displacement estimates by interpolating the given views from each other. Scene assumptions and knowledge of the camera configuration of the given views are used to fill-in displacement information for occluded regions. The complete and processed displacement vector field then can be used in a standard displacement-compensated interpolation algorithm to generate the desired view.

This enhancement has been used in the interpolation of numerous multi-view images. The effect of displacement estimation errors were significantly reduced and occluded regions were accurately synthesized, as evidenced by comparisons with actual intermediate views. Multiple interpolated views have been displayed on a multi-view binocular imaging system, with subjectively effective motion parallax.

1.5 Thesis Outline

We begin Chapter 2 by providing a formal definition of single-view video signals and extend this notion to multi-view systems. We then discuss the relative difficulty in coding multi-view signals with arbitrary viewpoints as compared to those with periodic camera parameters. Finally, we provide the quantitative cost measures and describe the multi-view video signals used in the analysis of the algorithms developed throughout the remainder of this thesis.

The theoretical basis for the multi-view compression algorithms developed in this thesis is presented in Chapter 3. We model image intensities in terms of a non-stationary, Gauss-Markov process. The non-stationarity is due to spatiotemporally-varying correlation coefficients, which model variations in lighting and camera pick-up parameters both within and between frames. We assume a rigid-body, translational displacement model for scene objects, and we make numerous observations on the geometrical constraints of displacement vector fields. Fundamental results are the relationship between displacement discontinuity and occlusions and the use of a displacement confidence measure to eliminate displacement estimates that contradict the established geometrical constraints.

In Chapter 4, we present our solution to the problem of selecting the reference frame that yields optimum prediction performance. We first provide motivation for the adaptive selection of the best possible reference frame. We then describe the selection algorithm in detail, which is based on the amount of occlusion between the desired and reference frames. This approach is compared with other possible reference frame selection methods, and the experimental results demonstrate the rate-distortion superiority of our algorithm.

We develop the novel class of restoration-based residual coders in Chapter 5. We begin by illustrating how the practical constraints on the frame-based prediction process result in a sub-optimal predicted image, which in turn can be used to improve the residual encoding process. We then describe the vector restoration coder in detail, which is very similar in implementation to a vector quantizer (VQ). We extend the common tree-structured, multi-stage, and variable rate VQ techniques to the basic restoration-based coder. This technique is compared with VQ and DCT-based methods through theoretical and experimental analysis.

In Chapter 6, we provide our method for accurately interpolating intermediate views from a decoded image pair by considering occluded and ambiguous regions. We briefly discuss the simple process of interpolation for unoccluded regions and develop assumptions that allow for range information of occluded regions to be inferred from adjacent unoccluded regions. We use numerous techniques developed in Chapter 4 to eliminate poor displacement estimates, and we refine the estimates further by interpolating the given image pair. The performance of this enhancement is evaluated through informal subjective assessments using a multi-view display system and through quantitative comparisons when the true intermediate views are available.

Conclusions and areas for future work are described in Chapter 7. We also discuss the relative contributions of these algorithms, and provide insight into how and when these techniques should be combined into a complete multi-view encoder/decoder system.

Chapter 2

Multi-view Video Signals

In this chapter, we extend the common notion of a video signal to include multiple, concurrent views of the scene. We then define arbitrary multi-view signals and illustrate the relative difficulty experienced in coding these signals compared to those with quasi-periodic camera parameters. Finally, we provide the quantitative measures used in the cost/benefit analysis of the algorithms developed, and describe the content and camera parameters of the multi-view signals used throughout this thesis.

2.1 Extension from Single-view Signals

A single-view video signal is generated by an optical sensor that samples the electromagnetic radiation of a scene on both a two-dimensional, spatial grid and also temporally. The three-dimensional single-view video signal (SVS) can be decomposed into a sequence of image frames, which are ideally the perspective projection of the visual information of the scene through the camera lens and onto the camera sensor at specific time instants. The notation for an SVS, consisting of N frames, is given by,

$$\mathrm{SVS} = F(nT), \; \forall n \in \{0, \ldots, N-1\} \tag{2.1}$$

where n is the frame index, T is the temporal sampling period, and $F(nT)$ is the n^{th} frame in the sequence. Each frame contains the complete set of samples captured on the image sensor plane, and is denoted by,

$$F(nT) = I(u, v, nT), \; \forall u \in \{0, \ldots, U-1\} \text{ and } \forall v \in \{0, \ldots, V-1\} \tag{2.2}$$

where $I(u, v, nT)$ is the picture element (pixel) at location (u, v) within the $U \times V$ pixel frame.

The content of the video signal is related to a set of parameters that uniquely describe the camera that generated the signal. These parameters provide the camera viewpoint, scale, and spectral bandwidth sensitivity. The variation of these parameters between temporal samples can provide knowledge of interframe correlation. For example, if the camera alters its viewpoint by panning across the scene, the displacement of all motionless objects between frames is constant over the entire frame and is directly proportional to the pan-angle. However, due to the relative difficulty in quantifying the parameter variations over time, most single-view compression techniques neglect the effect of these parameters; the camera parameters are assumed to be static and the evolution of the signal content is assumed to be the result of object motion.[1]

We now extend our notion of video signals to include signals obtained from the parallelization of multiple SVS's, which we refer to as multi-view video signals (MVS). The cameras that generated these distinct views obviously must have varied in at least one camera parameter. The relative camera parameters can be used to index the views and to provide knowledge of the inter-view correlation. Therefore, we will incorporate the relevant parameters into the notation of multi-view video signals. We assume that the relative values are constant throughout the duration of the video signal, and that the pertinent parameters are calibrated initially. These assumptions eliminate the difficult process of tracking the absolute camera parameters through time.

Since we have restricted ourselves to the compression of multi-view video signals with distinct viewpoints, we fix and, subsequently, disregard all other camera parameters in the notation for MVS's. For simplicity, we assume a pinhole camera lens model and that the camera axis is perpendicular to the plane of the camera sensor. Applying these constraints, the camera model and viewpoint are specified by three triplets that respectively denote: 1) the position of the camera lens, $\boldsymbol{c}$, 2) a point, $\boldsymbol{p}$, that lies on both the sensor plane and the camera axis, and 3) the endpoint, $\boldsymbol{e}$, of a unit vector that lies on the sensor plane, is parallel with the u-axis of the image sensor and whose origin is $\boldsymbol{p}$.[2] The line defined by points $\boldsymbol{p}$ and $\boldsymbol{c}$ forms the camera axis, which specifies camera pan and tilt. The focal length of the camera is $f = \|\boldsymbol{p} - \boldsymbol{c}\|$. The roll vector, given by $\boldsymbol{e} - \boldsymbol{p}$, provides the rotation of the camera about its axis. The relationship between these points and the mapping between 3-D world- and 2-D image-coordinates are illustrated in Fig. 2.1.

1. See [104] for a description of techniques that consider the effect of global camera motion.
2. It should be noted that the point $\boldsymbol{p}$ need not, and sometimes should not, lie at the center of the image sensor plane. Offsets from the center are useful in binocular imaging applications to provide geometrically-correct stereo-vision [33].

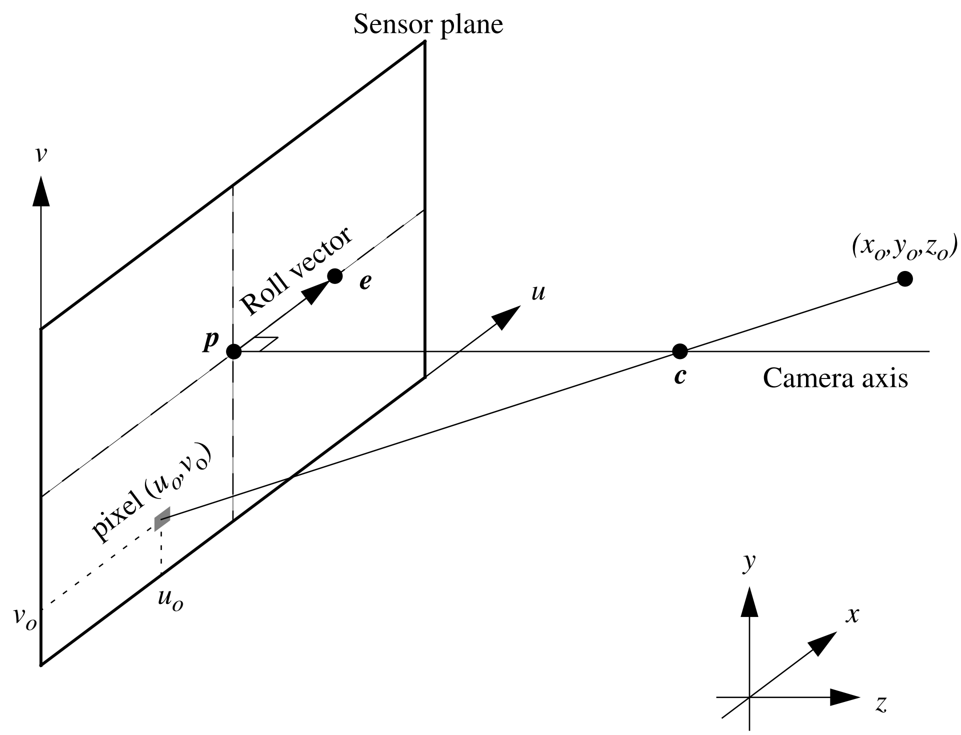


Figure 2.1: Camera viewpoint and relationship between world-coordinates and pixel location.

Due to the constraint on the camera axis being perpendicular to the sensor plane, only six parameters are needed to completely describe the camera viewpoint.[3] We distinguish each view within an MVS by these parameters and define the viewpoint matrix of the m^{th}-indexed view as,

$$\Phi_m = \begin{bmatrix} \boldsymbol{c} & \Theta \end{bmatrix} = \begin{bmatrix} c_x & \theta_x \\ c_y & \theta_y \\ c_z & \theta_z \end{bmatrix} \tag{2.3}$$

where θ_x, θ_y, and θ_z respectively denote the tilt, pan, and roll angles of the camera about its lens, and are related to the three triplets of the camera model.

Instead of denoting each view by its absolute viewpoint parameters, we specify a reference viewpoint matrix and express all other matrices in the MVS with respect to this reference. By convention, we use Φ_0 as the reference viewpoint matrix. We then define an orthonormal, right-handed coordinate system (R_{C_0}) for the 0^{th}-indexed view, whose origin is located at $\boldsymbol{c}_0$, positive z-axis is coincident with the camera axis pointing away from the sensor plane, and positive x-axis is parallel with, and in the direction of, the roll vector. Expressed in terms of this coordinate system, Φ_0 reduces to the null-matrix.

All other viewpoint matrices in the MVS are expressed in terms of Φ_0 by,

$$\Phi_m = \Phi_0 + \Delta\Phi_m \tag{2.4}$$

where $\Delta\Phi_m$ is the differential viewpoint matrix of the m^{th} view with respect to the 0^{th} view. The differential viewpoint matrices may be interpreted as the parameters necessary to convert a point expressed in the m^{th} camera's coordinate system to a point in R_{C_0}; the first column vector specifies the translation vector and the second column vector specifies the angles of the 3-D rotation matrix.

Finally, since it is often difficult to capture multi-view signals with synchronized temporal sampling, we allow for a possible phase offset between the individual views in our notation. We

3. These six parameters do not include the camera focal length, which is needed for the mapping from 3-D to 2-D coordinates. Again, since we have fixed the scale between views, we omit the focal length from our viewpoint parameter set.

define the scalar Δt_m as the relative temporal sampling offset of the m^{th}-indexed with respect to the 0^{th}-view, where $\Delta t_0 = 0$ and $|\Delta t_m| < T$. The complete notation for an MVS, with M views, then is obtained by extending Eq. (2.1) to include the differential viewpoints matrices and the phase offset:

$$\text{MVS} = F(\Delta\Phi_m, nT + \Delta t_m), \forall m \in \{0, \ldots, M\} \text{ and } \forall n \in \{0, \ldots, N-1\} \quad (2.5)$$

In a similar manner, individual frames within an MVS are denoted by incorporating these parameters into Eq. (2.2). For simplicity, we often will denote the individual frames within the multi-view signal by its viewpoint and temporal indices as $F(m, n)$. A fractional value for the temporal index (n) indicates the frame's phase offset; e.g., $F(1, 23.5)$ is the 23^{rd} frame in View 1 captured at $t = 23.5T$. The differential viewpoint matrices indicate the degree of inter-view correlation and the possible range of corresponding points with the various views.[4]

2.2 Arbitrary Viewpoints

In this thesis, we are concerned with the compression of multi-view video signals with *arbitrary* viewpoints. We next define arbitrary viewpoints and discuss our motivation for restricting our focus to this class of signals.

Arbitrary multi-view signals are most easily defined through their antonym: the class of multi-view signals that contain a subset of at least three views whose viewpoint parameters vary in a quasi-periodic manner. The differential viewpoint matrices for these non-arbitrary views are given by,

$$\Delta\Phi_m = q(m)\Delta\Phi \quad (2.6)$$

where $\Delta\Phi$ is the fundamental sampling period, and $q(m)$ is a scalar function that allows for an irregular periodic variation of the viewpoint parameters. In the simplest case, $q(m) = m$ and the

4. A complete derivation of the conversion between coordinate systems and the transformation from 3-D world- to 2-D image-coordinates can be found in [27].

viewpoints possess a strict periodicity. We define the two exterior views of periodic MVS's as the views where $q(m)$ is at an extremum; all other views are classified as interior views.

The majority of prior research on the compression of more than two views has dealt with non-arbitrary viewpoints with zero phase offset [7, 30, 40, 47]. These periodic signals require a very high camera complexity to generate the multitude of views, and they possess a simple relationship between the interior and exterior views. If the two exterior views have some overlap in their viewing areas and the interior views do not contain any regions occluded from both exterior views, the interior views contain completely redundant information that is readily extracted from the exterior views. We can best illustrate this relationship between the interior and exterior views of non-arbitrary multi-view signals by re-visiting our motion parallax application (Section 1.1.2).

Assuming that we only are interested in providing the viewer with simulated horizontal motion parallax, we require multiple, closely-spaced views whose differential viewpoint matrices are given by,

$$\Delta\Phi_m = m\begin{bmatrix} \Delta c_x & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.7}$$

Since the viewpoints only vary in their horizontal position, the displacement of corresponding points is restricted to lie along the u-axis of the image frame. A three-dimensional structure of twenty images, obtained from such a setup with a camera spacing of 5 mm, is shown in Fig. 2.2a. A horizontal slice through the middle of this structure is shown in Fig. 2.2b. This slice contains the middle row of each image and is commonly referred to as an epipolar plane image (EPI).

The EPI consists of a set of intersecting parallelograms, which represent the displacement of scene objects through the various views. An object becomes occluded when its parallelogram is intersected by another object's parallelogram. Since only two corner locations and the slope of each parallelogram are needed to completely describe the EPI from the given exterior views' rows, the interior rows can be encoded very efficiently. Further compression can be achieved, for the entire set of views, by exploiting the continuity relationship between neighboring epipolar plane images.

Other periodic variations of the viewpoint parameters result in similar structures with high degrees of redundancy. While the task of extracting the information to efficiently encode these structures is non-trivial, we feel that the high compression ratios often reported for non-arbitrary

a.

b.

Figure 2.2: Periodic, horizontally-spaced views, a) twenty stacked images, b) epipolar plane image (EPI) for middle row of stacked images.

viewpoints are obtained by essentially oversampling the scene [7, 30, 40, 47]. To avoid the generation and compression of the inherently redundant interior views, we wish to capture and encode only the exterior views and to accurately interpolate the interior views. Our desire to only use the exterior views is the basis of our focus on arbitrary multi-view signals.

This discussion on arbitrary viewpoints notwithstanding, we will assume that the viewpoints of the MVS that we wish to encode have sufficient overlap with one another to ensure a certain degree of correlation between views. This assumption is consistent with the notion of single-view video signals, where the typical frame rate of 30 Hz results in temporally-adjacent frames to be highly similar. While applications of multi-view signals with non-overlapping viewpoints can be construed, the degree of compression achieved by using the knowledge of all views available most likely will be equivalent to that which would be obtained if the views were compressed independently. Also, displacement information between views cannot be obtained for such a situation, and interpolation of the interior views becomes impossible.

2.3 Cost Measures

The underlying goal of any compression algorithm is the minimization of the costs associated with the transmission and/or storage of the signal through the exploitation of both redundant and irrelevant information. In video compression, we desire to reduce both the coded signal's transmission bit rate and the distortion of the reconstructed imagery from its original form. These costs obviously are inversely related and cannot be minimized simultaneously; for a particular compression technique, a reduction in the bit rate will cause an increase in the image distortion, and vice versa. It is customary to evaluate various compression techniques either by fixing one cost and comparing the free cost, or by comparing an overall cost function that weights the contribution of the rate and distortion. These weights are obtained from implicit knowledge of the relative importance of each cost.

In our cost/benefit analysis of the techniques developed in this thesis, we are interested in not only rate and distortion, but also the costs associated with the complexity and storage requirements of each algorithm. Complexity and storage costs are valuable for evaluating the realizability of an algorithm; e.g., if a technique can achieve some optimum rate-distortion performance but it requires an excessive amount of time to yield this result, the algorithm has no practical use.

The overall cost function that we will use to evaluate techniques for the compression of multi-view signals is a linear combination of the coded bit rate (R), image distortion (D), algorithm complexity (C), and memory storage requirement (S), and is given by,

$$J = R + \lambda_D D + \lambda_C C + \lambda_S S \tag{2.8}$$

where each cost is weighted by its respective λ_i, with $\lambda_R = 1$ by our convention.[5] To facilitate comparison, we will often fix either the rate or distortion of a particular multi-view signal for each technique analyzed and examine the other costs. We next provide our definitions for the rate, distortion, complexity, and storage costs.

2.3.1 Bit Rate

Bit rate is defined simply as the average number of bits per second needed to represent the compressed video signal at its temporal frame rate. We also will use the number of bits per pixel and bits per frame as measures of coded bit rate.

2.3.2 Distortion

Distortion is inversely related to the fidelity, or quality, of the decoded images. By definition of lossy compression techniques, some form of distortion between the original and reconstructed images will always be present. The evaluation of distortion should be related to the application of the imagery; for visual communication systems, distortion ideally would be related to the perceptual capability of the human visual system. However, we know of no widely-accepted distortion measure that provides a quantitative result that coincides with subjective image quality. We will thus use the often-criticized class of distortion measures that calculate the L_p-norm between the intensity values of the original and reconstructed images [42]. These measures are easily calculated and allow for direct comparisons with other research results.

In its most basic form, we define distortion as,

$$D = \frac{1}{A} \sum_{\forall u, v \in A} \left| I(u, v) - \bar{I}(u, v) \right|^p \tag{2.9}$$

5. Since the relative values of these weights are system dependent, we leave their assignment to system designers.

where $\bar{I}(u, v)$ is the pixel intensity of the decoded image and A is the image area in pixels. Common variations of this distortion measure are the mean-absolute-difference (MAD, $p = 1$) and the mean-square-error (MSE, $p = 2$). While we have presented distortion as a measure over the entire image frame, we will often calculate the distortion of an image region by limiting the area over which the summing takes place.

The literature often quotes the signal-to-noise ratio (SNR) as a measure of image quality. The SNR of a reconstructed image is given by,

$$\mathrm{SNR\ (dB)} = 10 \log_{10}\left\{\frac{\sigma^2}{\mathrm{MSE}}\right\} \tag{2.10}$$

$$\sigma^2 = \left\{\frac{1}{A}\sum_{\forall u, v \in A} [I(u, v)]^2\right\} - \mu^2 \tag{2.11}$$

$$\mu = \frac{1}{A}\sum_{\forall u, v \in A} I(u, v) \tag{2.12}$$

where σ^2 and μ respectively denote the variance and mean of the original image's intensity.

The final distortion measure that we will use is referred to as the peak-signal-to-noise ratio (PSNR), which is obtained by substituting the peak-to-peak intensity range (e.g., 255 for 8-bit images) for σ in Eq. (2.10).

2.3.3 Complexity

The complexity of an algorithm is defined as the number of both fixed- and floating-point operations required to perform its functions. Due to the difficulty in exactly calculating these quantities, we will mainly be interested in the portions of an algorithm that require looping and repeated manipulation of image data. We will express complexity in terms of the number of operations per pixel, and we will provide complexity costs for both the encoder and decoder, since the algorithms presented are not symmetric.

2.3.4 Storage

The storage cost is defined as the number of bytes needed to store the data sets associated with an algorithm. This includes both frame memory and any other relatively large structures, such as look-up tables and codebooks, used in the compression of the image data. Again, due to the asymmetric nature of the algorithms examined, we will provide storage costs for both encoder and decoder structures.

2.4 Multi-view Sequence Descriptions

This section describes the multi-view video sequences used to analyze the performance of the various algorithms developed in this thesis. We next provide the camera parameters and a brief overview of the content of these signals. Each sequence was captured in YUV 4:2:0 format with 8-bit accuracy for each pixel component; the luminance and two color components were treated separately and the color components were subsampled by a factor of two in both the horizontal and vertical image directions. Unless specified, all sequences were captured by the author. The sequence highlights are included in Table 2.1.

The camera parameter sets labelled A and B describe the camera configurations and temporal sampling structures of the various sequences, and are illustrated in Figs. 2.3 and 2.4, respectively. The viewpoint index for each view corresponds to the appropriate camera number in these figures. The multiple time axes in these diagrams represent the temporal sampling of the scene for each view within the MVS. The black squares are the relative temporal locations of frames, and the dashed-lines that intersect the time axes indicate constant temporal values.

2.4.1 *Cement Mixer* Sequence

The *Cement Mixer*[6] sequence was obtained using three, horizontally-spaced cameras with synchronized temporal samples. The focal lengths of both cameras were 50 mm, and the camera spacing was fixed at 65 mm, resulting in the differential viewpoint matrices:

$$\Delta\Phi_m = m\begin{bmatrix}\Delta c_x & 0\\ 0 & 0\\ 0 & 0\end{bmatrix}, m = 0, 1, 2 \tag{2.13}$$

While these viewpoints meet our criteria for periodic multi-view signals, we use this sequence since the three views have a relatively large spacing that precludes the interior view from being

6. The image frames in this sequence were manually shifted and cropped once per-sequence to ensure a reasonable amount of overlap between inter-view frames and subjectively-pleasing stereoscopy.

Sequence Name	Number of Views	Parameter Set	Duration (seconds)	Frame Rate (Hz)	Frame Resolution
Cement Mixer	3	A	7	30	544 x 480
Volleyball	2	B	8	30	640 x 240
Finish Line	2	B	9	30	640 x 240
Skater	2	B	10	30	640 x 240
Toy Train	2	B	6	50	720 x 288
Piano	2	B	2	50	720 x 288
Manege	2	B	1	50	720 x 288
Flower Garden	1	N/A	5	30	352 x 240

Table 2.1: Sequence highlights

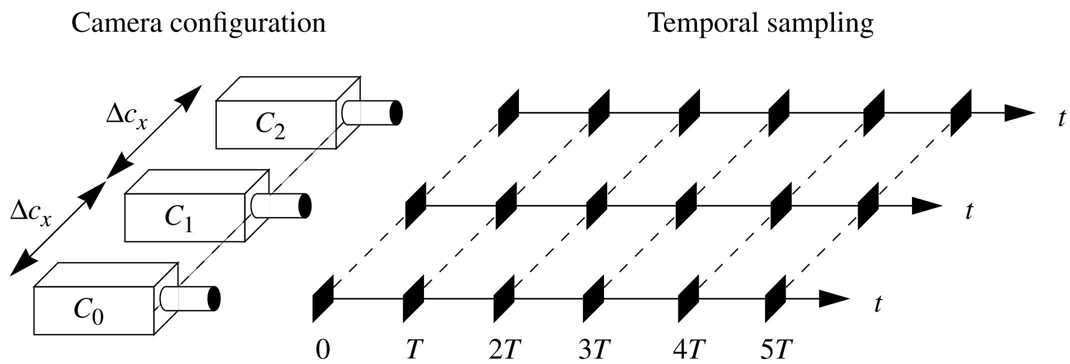


Figure 2.3: Camera Parameter Set A.

entirely redundant. The camera configuration and temporal sampling structure for this signal are shown in Fig. 2.3.

The content of this sequence includes a foreground tree, two parked cars, and a large cement truck in the background. The cameras exhibit no camera motion. The majority of the motion in the sequence is the slow rotation of the barrel of the cement truck, which contains lettering. Two pedestrians walk across the scene from right-to-left. Spatially-adjacent views possess approximately 15% occlusion.

The foci of the individual views were poorly aligned, resulting in a large variation in the object plane that was in focus for the frames within View 0 compared to the other two views. A sample frame from each of the three views is shown in Fig. 2.5.

2.4.2 *Volleyball* Sequence

The *Volleyball* sequence consists of two views obtained by horizontally-displaced cameras with parallel axes and coplanar image sensors. The viewpoint spacing was fixed at 160 mm and each camera had a focal length of 50 mm, such that:

$$\Delta\Phi_1 = \begin{bmatrix} \Delta c_x & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.14}$$

The temporal sampling of the two cameras was synchronized such that $\Delta t_1 = 0$. The parameter set for this sequence is shown in Fig. 2.4.

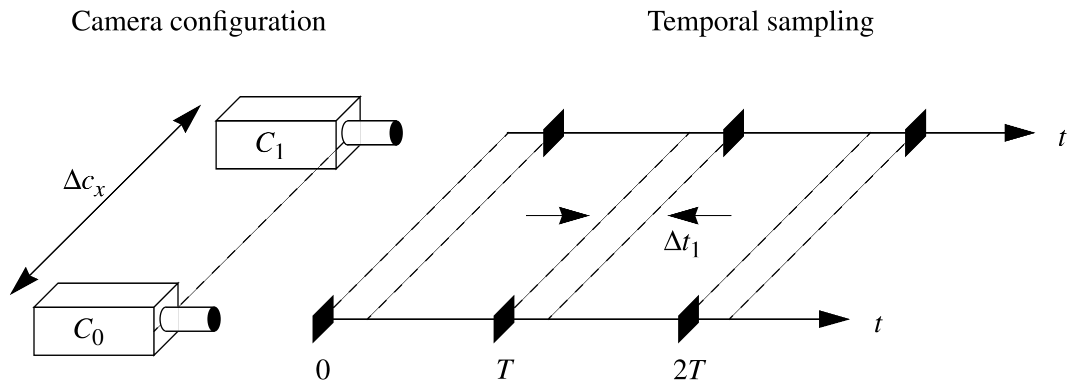


Figure 2.4: Camera Parameter Set B.

The scene is composed of a group of individuals engaged in a game of volleyball on a background of grass. The majority of the objects in the scene are located quite a distance away from the camera pair, resulting in a relatively small amount of inter-view occlusion. The camera pair exhibits random motion as the game progresses. Two frames from this sequence are provided in Fig. 2.6.

2.4.3 *Finish Line* Sequence

The *Finish Line* sequence was captured using a commercial binocular-camera. The original camera configuration was modified to ensure parallel camera axes and coplanar image sensors. The two viewpoints differed by a constant offset of 50 mm along the x-axis. The separate views were recorded using the field sequential characteristic of the camera, which resulted in a phase offset of one-half the frame period ($\Delta t_1 = \frac{T}{2} = \frac{1}{60}$ second) between identically-indexed frames within the two views. The camera configuration and temporal sampling structure of this signal are illustrated in Fig. 2.4

The scene is composed of a crowd watching the end of an outdoor racing event. The people in the scene exhibit moderate motion and the camera performs a slow, bidirectional pan. Although the viewpoints vary by only 50 mm along and the cameras had a focal length of 9.5 mm, the proximity of the people in the scene to the cameras results in a large degree of inter-view occlusion. A sample frame from each view is shown in Fig. 2.8, where the images have been scaled by one-half in the horizontal direction to ensure a realistic aspect ratio.

2.4.4 *Skater* Sequence

The identical camera setup of the previous sequence was used to capture the *Skater* two-view signal. The sequence is composed of a quadrangle of university buildings in the background. The camera pair follows the motion of an in-line skater as he traverses horizontally across the scene. The cameras exhibit a rapid pan resulting in a moderate degree of intra-view occlusion on the image frame borders. The pan rate varies greatly throughout the sequence and eventually the cameras stop moving. The 175th frame from each view is shown in Fig. 2.8.

2.4.5 *Toy Train* Sequence

The *Toy Train*[7] sequence was captured using a binocular camera with the same basic camera configuration and temporal sampling structure as shown in Fig. 2.4. The cameras were separated by a horizontal distance of 87.5 mm, had focal lengths of 40 mm, and a zero phase offset.

The scene consists of a model representing a mountain and an area containing houses, train tracks, figurines, cars, and associated objects. This sequences contains two distinct sections (commonly referred to as *Train* and *Tunnel* in the literature), with an abrupt change in the absolute camera viewpoints between sections. In the first portion of the sequence, two trains pass one another in opposite directions, and the cameras execute a slight pan. In the second section, a single train runs along a curved track and slowly enters a tunnel. A sample pair of views from the second section of this sequence is shown in Fig. 2.9.

2.4.6 *Piano* Sequence

The two-view *Piano*[7] sequence was captured using the same camera parameters and temporal sampling of the *Toy Train* sequence (Section 2.4.5). The scene consists of a man playing a piano, with a uniform blue-colored wall background. The body of the man undergoes a slight translation and his fingers exhibit complex motion. The cameras perform a slight pan. The fiftieth frame from each view of this sequence are illustrated in Fig. 2.10.

---

7. Courtesy of Centre Commun D’Études de Télédiffusion et Télécommunications (CCETT) and the DISTMA project. This set of sequences is regarded as standard in the stereoscopic sequence compression community.

2.4.7 *Manege* Sequence

The *Manege*[7] sequence was captured using the same basic camera configuration and temporal sampling as the previous two sequences, with a focal length of 30 mm. The horizontal spacing between viewpoints was increased to 100 mm. The scene consists of a moving carousel in the main square of a city. The complex motion of the objects on the carousel results in a large degree of occlusion both between views and between temporally-offset frames within a single view. The background contains some buildings and a road with moving vehicles. Sample frames from each view are depicted in Fig. 2.11.

2.4.8 *Flower Garden* Sequence

The *Flower Garden*[8] sequence is actually a single-view signal. Since the scene objects are static and the camera motion consists of a relatively uniform translation in the horizontal direction, we simulate a multi-view signal by using temporally-offset copies of this sequence.

The scene consists of a gently-sloping hill that contains a flower garden and a large foreground tree. Due to the relative closeness of the foreground tree, a high degree of occlusion is present in the portion of the sequence that contains this object. A building is located in the background, on top of the hill. Two temporally-offset frames from this sequence are provided in Fig. 2.8.

8. Courtesy of Rensselaer Polytechnic Institute (RPI), ftp://ftp.ipl.rpi.edu.

View 0

View 1

View 2

Figure 2.5: *Cement Mixer* sequence, frame 50.

View 0

View 1

Figure 2.6: *Volleyball* sequence, frame 200.

View 0

View 1

Figure 2.7: *Finish Line* sequence, frame 100.

View 0

View 1

Figure 2.8: *Skater* sequence, frame 175.

View 0

View 1

Figure 2.9: *Toy Train* sequence, frame 150.

View 0

View 1

Figure 2.10: *Piano* sequence, frame 50.

View 0

View 1

Figure 2.11: *Manege* sequence, frame 25.

Frame 75

Frame 77

Figure 2.12: *Flower Garden* sequence, frames 75 and 77.

Chapter 3

Theoretical Basis for New Multi-view Algorithms

We present two models that form the theoretical basis for the multi-view compression algorithms developed in the subsequent chapters of this thesis.

A non-stationary, Gauss-Markov process is used to model pixel intensities both within and between image frames. The non-stationarity is due to the use of spatiotemporally-varying correlation coefficients that model the realistic variations in lighting and camera pick-up parameters between pixels. Intraframe and interframe forms of the model are provided, which respectively describe occluded and unoccluded image regions. From this model, we illustrate why the reference frame with the minimal amount of occlusion should be used in the frame-based prediction process, which we address in Chapter 4. We also allude to the sub-optimal nature of the predicted image due to bit-rate constraints – this concept is explored in depth in Chapter 5.

An integral part of the image model for unoccluded regions is the displacement of corresponding points. We use the conventional rigid-body, translational displacement model to describe the relative shift of image regions between frames. We then develop a few fundamental principles related to this displacement model and block-based displacement estimation techniques, including: 1) the one-to-one mapping of bidirectional displacement vectors, 2) the relationship between occlusion and displacement discontinuity, and 3) the use of the prediction block-SNR as a reliability measure for estimated displacement vectors. These results are used extensively in the algorithms presented in Chapters 4 and 6.

3.1 Image Model

Since we are concerned with interframe predictive coding, we distinguish image regions within a frame as being either unoccluded or occluded with respect to another frame in the multi-view signal. We label the two frames under consideration as the *current* and *reference* frames, where the current frame is to be encoded using knowledge of the reference frame. Unoccluded regions in the current frame are those portions that are “visible” in the reference frame, i.e., a corresponding region is present in the reference frame. Conversely, image regions in the current frame that are “hidden” from the reference frame are classified as being occluded. Due to the varying statistical characteristics of each of these regions, we present two distinct, yet related, models for pixel intensities.

3.1.1 Unoccluded Regions

The discrete-time, Gauss-Markov process has been shown to adequately model image sequence intensities along motion trajectories [28, 29]. Since the displacement of an image region between two frames within a multi-view signal is analogous to an object’s motion trajectory between two temporally-offset frames, we adopt this model for our purposes. For mathematical tractability and to facilitate theoretical comparisons of coding schemes, the Gauss-Markov process typically is assumed to be asymptotically stationary; however, we feel this simplification does not represent the true nature of video signals.

The intensity of corresponding points may vary between frames due to specular reflections, lighting variations, or changes in camera pick-up parameters (e.g., focus, brightness, contrast, and color balance). These situations are more prevalent in multi-view signals due to the distinct camera viewpoints and the difficulty in obtaining exact camera calibration. The additive, white-noise component in the stationary Gauss-Markov process does not adequately model these variations. Therefore, we include a multiplicative term, in the form of a spatiotemporally-varying correlation coefficient, to the conventional image model.

We denote the current and reference frames by $F(m, n)$ and $F(m_{\text{ref}}, n_{\text{ref}})$, respectively. The reference frame may be offset from the current frame in time and/or viewpoint. The 1$^{\text{st}}$-order, non-stationary Gauss-Markov model of the image intensity for an unoccluded pixel in the current frame, along its displacement trajectory, then is given by,

$$I(u, v, m, n) = a_{00}(u, v) \cdot I(u - d_u(u, v), v - d_v(u, v), m_{\text{ref}}, n_{\text{ref}}) + w_{\text{un}}(u, v) \qquad (3.1)$$

where d_u and d_v respectively denote the horizontal and vertical image components of the pixel’s displacement (see Section 3.2), $w_{\text{un}}(u, v)$ is the i.i.d. Gaussian innovations sequence, and $a_{00}(u, v)$ is the correlation coefficient. While each of these quantities is related to the specific current and reference frames under consideration, we have omitted this dependence from our notation for clarity.

The innovations sequence is a white-noise random process independent of previous intensity values along the displacement trajectory. This model reduces to the stationary case when the correlation coefficient is fixed. Typically, video sources are assumed to be highly correlated with coefficient values in the range 0.9 to 1.0 [29]. Due to the possibility of pixel-by-pixel intensity fluctuations, we make no assumptions on the rate of change of the correlation coefficient.

While this more complex model precludes the development of closed-form solutions for the rate-distortion function of this source, we feel that it more accurately represents the true nature of image intensities through time and space.

3.1.2 Occluded Regions

Since displacement is defined only for image regions visible in both the current and reference frames, we cannot model occlusions using Eq. (3.1). Instead, we represent the intensity of an occluded pixel in terms of adjacent pixels within the current image frame, i.e., an intraframe representation. For consistency, we again adopt a non-stationary, Gauss-Markov model, given by,

$$I(u, v, m, n) = \sum_{\forall i, j \in \aleph_p} a_{ij}(u, v) \cdot I(u-i, v-j, m, n) + w_{\text{occ}}(u, v) \tag{3.2}$$

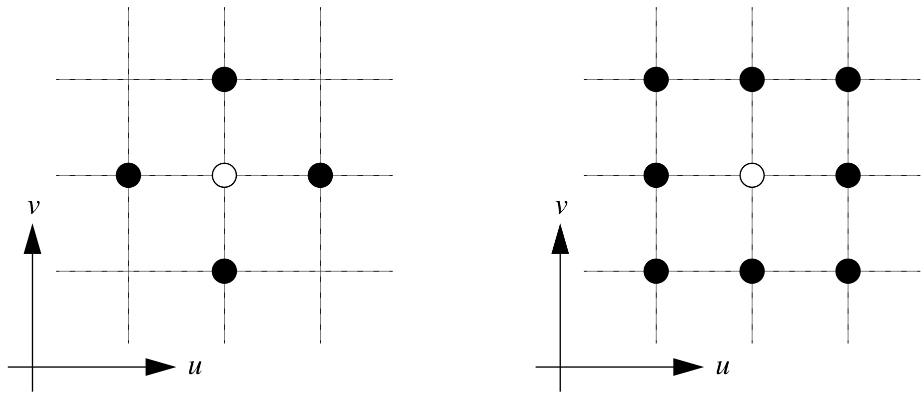
with innovations sequence $w_{\text{occ}}(u, v)$, and correlation coefficients $a_{ij}(u, v)$ defined over the neighborhood $\aleph_p$.

To maintain the greatest generality, we use a noncausal neighborhood of adjacent pixels.[1] This neighborhood is defined as the set of p-closest pixels, in the sense of euclidean distance, from the pixel under consideration. The pixel locations for the $\aleph_4$ and $\aleph_8$ neighborhoods are depicted by the blackened circles in Fig. 3.1, where the hollow circle represents $I(u, v, m, n)$.

3.1.3 Compact Representation of Image Model

The inherent redundancy of the source signal, in the form of memory, is evident from Eqs. (3.1) and (3.2). A predictive coder attempts to exploit this memory to provide a more concise represen-

1. See [8] and [43] for a complete description of noncausal models.



a. b.

Figure 3.1: $\aleph_4$ (a) and $\aleph_8$ (b) two-dimensional, noncausal neighborhoods.

tation of the multi-view signal. We next consider the task of generating a compact representation for unoccluded regions.

The best linear mean square prediction of an unoccluded pixel, from a given reference frame, is provided by the combination of correlation coefficient and displacement vector. If these parameters are specified exactly, the error, or residual, between the actual and predicted pixel intensities will equal the innovations sequence. Typically, the entropy of the residual image is considerably less than that of the original image; hence, it can be encoded using fewer bits for a given level of distortion. However, an excessive number of bits would be required to transmit the correlation coefficient and displacement vector for each pixel. Since these quantities represent an over-determined set of parameters for the representation of the desired pixel, approximations of the image model are often made to reduce the bit rate required to describe the prediction process.

Even though we hypothesize that our non-stationary model more accurately represents the true progression of image intensities through a video signal, parameter estimation of the correlation coefficient is a difficult process. This difficulty is due not only to the spatiotemporal variation of the coefficient, but also to the fact that the actual displacement vectors are unknown and must be estimated themselves. Therefore, one cannot directly calculate the covariance of the image intensities along displacement trajectories, which is required to estimate the best linear mean square prediction filter. A common approximation then is through the use of a spatially-invariant prediction filter, i.e., the filter is fixed over an entire image. Since the correlation coefficient of the image model is sufficiently close to one, a unity filter tap is often employed. Although this prediction filter has a pole on the unit-circle, stability is not an issue since the output is bounded by the quantization of intensity values.

Another often used approximation is that contiguous regions of pixels have constant displacement. This approximation is based on a rigid-body, translation displacement model. Only a single displacement vector must be transmitted for each group of pixels, greatly reducing the bit rate of the prediction.

Combining these approximations, the prediction of the pixel in the current frame, from the corresponding point in the given reference frame is given by,

$$\hat{I}(u, v, m, n) = I(u - \hat{d}_u(u, v), v - \hat{d}_v(u, v), m_{\text{ref}}, n_{\text{ref}}) \tag{3.3}$$

where $\hat{d}_u$ and $\hat{d}_v$ are interpreted as the average horizontal and vertical displacement components of the image region assumed to possess constant displacement. This prediction is sub-optimal in the sense that the error between the actual and predicted intensities no longer equals the innovations sequence. Instead, the residual is correlated with the prediction, and generally requires more bits to encode than that required for $w_{\text{un}}(u, v)$. Therefore, a trade-off exists between allocating more bits to accurately represent the parameters of the image model or more bits to encode the residual image. We discuss the ramifications of the predicted-residual image correlation and a technique to exploit this information in Chapter 5.

A similar process is performed for the compression of occluded regions, where a single set of correlation coefficients typically are assumed or estimated for the entire image frame. We note a few observations on techniques used to obtain a compact representation for occluded regions.

The noncausal model lends itself to transform-based techniques [43]. This is the approach taken by the MPEG class of video compression standards [2], where intraframe DCT-based compression is performed on “poorly predicted” (read occluded) blocks within a predictively-coded frame. The use of a fixed transform implies the assumption of a stationary image model. This concept was extended in [24] to include image dependent statistics; an image block is coded using a transform selected from a codebook of transforms designed off-line, each of which approximates the KLT for a specific class of image sources. This approach can accommodate a finite number of correlation variations.

Noncausal prediction, as described in [6], also can be used for the intraframe coding of occlusions. Here, horizontal and vertical sample correlations are calculated over the entire image, and are transmitted to the decoder. The image then is “whitened” using a backward representation of the potential matrix, and the residual is encoded via a vector quantizer. The complexity of this technique can be reduced, from its original form, since occlusions typically occupy only a relatively small percentage of the image frame, and boundary values are readily available from neighboring unoccluded regions.

Regardless of the method taken to generate a compact representation of occluded regions, in general, substantially more bits are required to encode these regions as compared to unoccluded regions; i.e., the entropy of $w_{\mathrm{occ}}(u, v)$ is greater than that of $w_{\mathrm{un}}(u, v)$. This discrepancy in rate-distortion performance typically is most severe for highly-textured image regions, since the intraframe model for these regions yields a poor prediction of the desired pixel intensity, in terms of a large dynamic range for the prediction error.

Therefore, we wish to use the reference frame that has minimum occlusion with the current frame for the frame-based prediction step of a multi-view hybrid encoder. We provide our solution to the selection of the minimum-occlusion reference frame in Chapter 4.

3.2 Displacement Model

Displacement is formally defined as the vector difference between the pixel locations of corresponding points in appropriately aligned image frames. Displacement is a function of scene structure, differential viewpoint parameters, object motion and temporal sampling rate.

The displacement vector for an unoccluded pixel in Frame A, with respect to reference Frame B, is denoted by,

$$\boldsymbol{d}_{A \leftarrow B}(u, v) = \begin{bmatrix} d_{u_{A \leftarrow B}}(u, v) \\ d_{v_{A \leftarrow B}}(u, v) \end{bmatrix} \tag{3.4}$$

If there is no ambiguity, we will often omit the subscripts from our displacement notation. Since displacement is undefined for occluded pixels, we set $\boldsymbol{d}(u, v) \equiv \varnothing$ for these regions. We define the displacement vector field (DVF) as the collection of displacement vectors over the entire image frame:

$$\mathrm{DVF} = \boldsymbol{d}(u, v),\ \forall u \in \{0, \ldots, U-1\} \text{ and } \forall v \in \{0, \ldots, V-1\} \tag{3.5}$$

A useful quantity is the displacement vector discontinuity, which is defined as the difference between displacement components of unoccluded pixels along the two image coordinates:

$$\Delta \boldsymbol{d}(u, v) = \begin{bmatrix} \Delta d_u(u, v) \\ \Delta d_v(u, v) \end{bmatrix} = \begin{bmatrix} d_u(u, v) - d_u(u-1, v) \\ d_v(u, v) - d_v(u, v-1) \end{bmatrix} \tag{3.6}$$

where $d_u(-1, v) \equiv 0$ and $d_v(u, -1) \equiv 0$. These difference equations can be viewed as the partial derivatives of the displacement along the *u*- and *v*-coordinates. We also use the form of Eq. (3.6) for the discontinuity between adjacent image regions, when pixels within each region have constant displacement.

When dealing with estimated displacement vectors, we will include a circumflex into this notation, i.e., $\hat{\boldsymbol{d}}(u, v)$ and $\Delta\hat{\boldsymbol{d}}(u, v)$ respectively denote the estimated displacement and estimated displacement discontinuity. Displacement estimates that are assumed to be inaccurate, based on some knowledge of the reliability of the estimate, are likewise denoted by $\hat{\boldsymbol{d}}(u, v) = \varnothing$.

The complete estimated displacement vector field uniquely describes the frame-based prediction of the current frame from Eq. (3.3). Fractional displacement estimates are handled in Eq. (3.3) by performing some form of spatial interpolation to generate the intensity value at sub-pixel locations in the reference frame.

For simplicity, we assume a rigid-body, translational displacement model throughout this thesis, i.e., a contiguous region of pixels has constant displacement. Or, in terms of the scene, we assume that the scene consists of a collection of "small" planar surfaces, which are perpendicular to the camera axis and exhibit only planar motion between the temporal samples of the current and reference frames. This model represents the relatively complex, affine geometric transformations of rotation and scaling as a set of piecewise linear translations.

The displacement component along its corresponding image coordinate is a staircase function; i.e., it consists of a set of flat regions with distinct jumps at the boundaries between objects either located at different depth planes (for spatially-offset frames), or possessing differential velocities (for temporally-offset frames). The displacement discontinuity then is a series of Kronecker delta functions located at object boundaries.[2]

2. In statistical terms, the displacement along an image row (column) can be modeled as a generalized Poisson process, and the displacement discontinuity as a generalized Poisson impulse process [73], where the location of the Poisson points are correlated between adjacent image rows (columns).

We next develop two fundamental principles based on the rigid-body, translation displacement model. We follow this by describing a reliability measure for block-based displacement estimates. These results will be used extensively in Chapters 4 and 6 for the processing of noisy displacement vector fields.

3.2.1 One-to-One Mapping of Unoccluded Regions

Consider an image region, $\Re_A$, in Frame A that is visible in reference Frame B at the corresponding region denoted by $\Re_B$. We describe this relationship by stating that, "$\Re_B$ maps to $\Re_A$". Obviously, the reverse relationship holds: $\Re_B$ is visible in Frame A at $\Re_A$. This bidirectional relationship represents the one-to-one mapping between unoccluded regions.

Neglecting image noise and lighting variations from Eq. (3.1), the intensity of an unoccluded pixel in Frame A is expressed in terms of the intensity of the corresponding point in Frame B from Eq. (3.7a), and the mapping of corresponding pixels from Frame A to Frame B is given by Eq (3.7b).

$$I(u, v, A) = I(u - d_{u_{A \leftarrow B}}(u, v), v - d_{v_{A \leftarrow B}}(u, v), B) \tag{3.7a}$$

$$I(u, v, B) = I(u - d_{u_{B \leftarrow A}}(u, v), v - d_{v_{B \leftarrow A}}(u, v), A) \tag{3.7b}$$

We define the change of variables, $u' = u - d_{u_{A \leftarrow B}}(u, v)$ and $v' = v - d_{v_{A \leftarrow B}}(u, v)$. Substituting these quantities and reversing the left-hand and right-hand sides of Eq. (3.7a) yields,

$$I(u', v', B) = I(u' + d_{u_{A \leftarrow B}}(u, v), v' + d_{v_{A \leftarrow B}}(u, v), A) \tag{3.8}$$

Equating Eq. (3.8) with Eq. (3.7b), we obtain the relationship between bidirectional displacement vectors for unoccluded regions:

$$\boldsymbol{d}_{B \leftarrow A}(u - d_{u_{A \leftarrow B}}(u, v), v - d_{v_{A \leftarrow B}}(u, v)) = -\boldsymbol{d}_{A \leftarrow B}(u, v) \tag{3.9}$$

This result implies that the displacement vector field relating Frame B from Frame A is uniquely specified from the DVF of Frame A from Frame B. Therefore, given a unidirectional displacement field, the complete reversed field is obtained by calculating the displacement vector for each pixel using Eq. (3.9). We may also use this relationship to disregard displacement estimates that are inconsistent with the one-to-one mapping. A final observation is that the number of occluded pixels in Frame *A*, with respect to Frame *B*, equals the number of occluded pixels in Frame *B*, with respect to Frame *A*.

3.2.2 Occlusion / Displacement Discontinuity Relationship

We establish the fundamental property relating occlusion and displacement discontinuity by examining the source of occluded regions.

Assume that a real-world object lies completely within a camera's field-of-view and is located between the camera and some scene background. An occlusion is caused by the foreground object if and only if the camera viewpoint changes or the object moves with respect to the background between frames. For either case, the displacement of the foreground object differs from that of the scene background, i.e., a displacement discontinuity exists.

We demonstrate the occlusion / displacement relationship by examining the effect of a displacement discontinuity in a unidirectional field on the reversed field. Let the horizontal displacement components of two adjacent unoccluded pixels within $\mathrm{DVF}_{A \leftarrow B}$ equal d_1 and d_2, respectively:

$$d_{u_{A \leftarrow B}}(u-1, v) = d_1 \text{ and } d_{u_{A \leftarrow B}}(u, v) = d_2 \tag{3.10}$$

To simplify the notation, we assume that $d_{v_{A \leftarrow B}}(u-1, v) = d_{v_{A \leftarrow B}}(u, v) = 0$. Substituting these quantities into Eq. (3.9) yields the horizontal displacements for the two corresponding points in the reversed field:

$$d_{u_{B \leftarrow A}}(u-1-d_1, v) = -d_1 = d_{u_{B \leftarrow A}}(u_1, v) \tag{3.11a}$$

$$d_{u_{B \leftarrow A}}(u-d_2, v) = -d_2 = d_{u_{B \leftarrow A}}(u_2, v) \tag{3.11b}$$

If $d_1 = d_2$, then $u_1 = u_2 - 1$. However, if a discontinuity exists at this location, the two adjacent pixels in Frame *A* are mapped from non-adjacent pixels in Frame *B*. While the region between (u_1, u_2) in Frame *B* might map to another region in Frame A, it is more likely that this region is occluded, where the horizontal length of the occlusion equals $|d_1 - d_2|$.

We extend this relationship to two-dimensions with the illustration shown in Fig. 3.2a, where two 58×23 pixel image frames are drawn in perspective. Each image frame contains a lightly-shaded, 16×8 pixel foreground object and a textured background. Between frames, both the foreground object and the background exhibit purely translation displacement, resulting in constant displacement for all pixels within each image region. The actual displacements from Frame *A* to Frame *B* for the entire foreground object and background are, respectively,

$$\boldsymbol{d}_{\text{fore}} = \begin{bmatrix} 7 \\ -3 \end{bmatrix} \text{ and } \boldsymbol{d}_{\text{back}} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \tag{3.12}$$

We have indicated the occlusions within each frame with the checkerboard regions, where the alternating black/transparent rectangles represent image pixels.

Graphically, it is evident that the size of an occlusion due to the uniform displacement of an object with respect to its background is,

$$\text{occlusion size} = (\text{object area}) - (\text{self-overlap area}) \tag{3.13}$$

where “self-overlap area” is the size (in pixels) of the object’s overlap with itself in appropriately-aligned backgrounds between frames. The overlap area is a function of displacement discontinuity, is non-negative, and is less than or equal to the object area. Therefore, the maximum occlusion size is equal to the object area.

For a rectangular region of $w \times h$ pixels, Eq. (3.13) can be expressed as,

$$\text{occlusion size} = w \cdot h - \{\max(0, w - |\Delta d_u|) \cdot \max(0, h - |\Delta d_v|)\} \tag{3.14}$$

Obviously, if the displacement discontinuity vector between two regions is the null-vector, no occlusion exists. In our diagram, the displacement discontinuity of the foreground object with respect to the background object is $\Delta\boldsymbol{d}_{\text{fore-back}} = \begin{bmatrix} 7 & -5 \end{bmatrix}^t$, and Eq. (3.14) yields the correct result of 101 pixels. The same calculation can be made for the occlusion due to the constant displacement of the background by realizing that the background itself can be viewed as lying on a zero-displacement background. This yields an occlusion size of $2U = 116$ due to the background, for a total of 217 occluded pixels in each image frame.

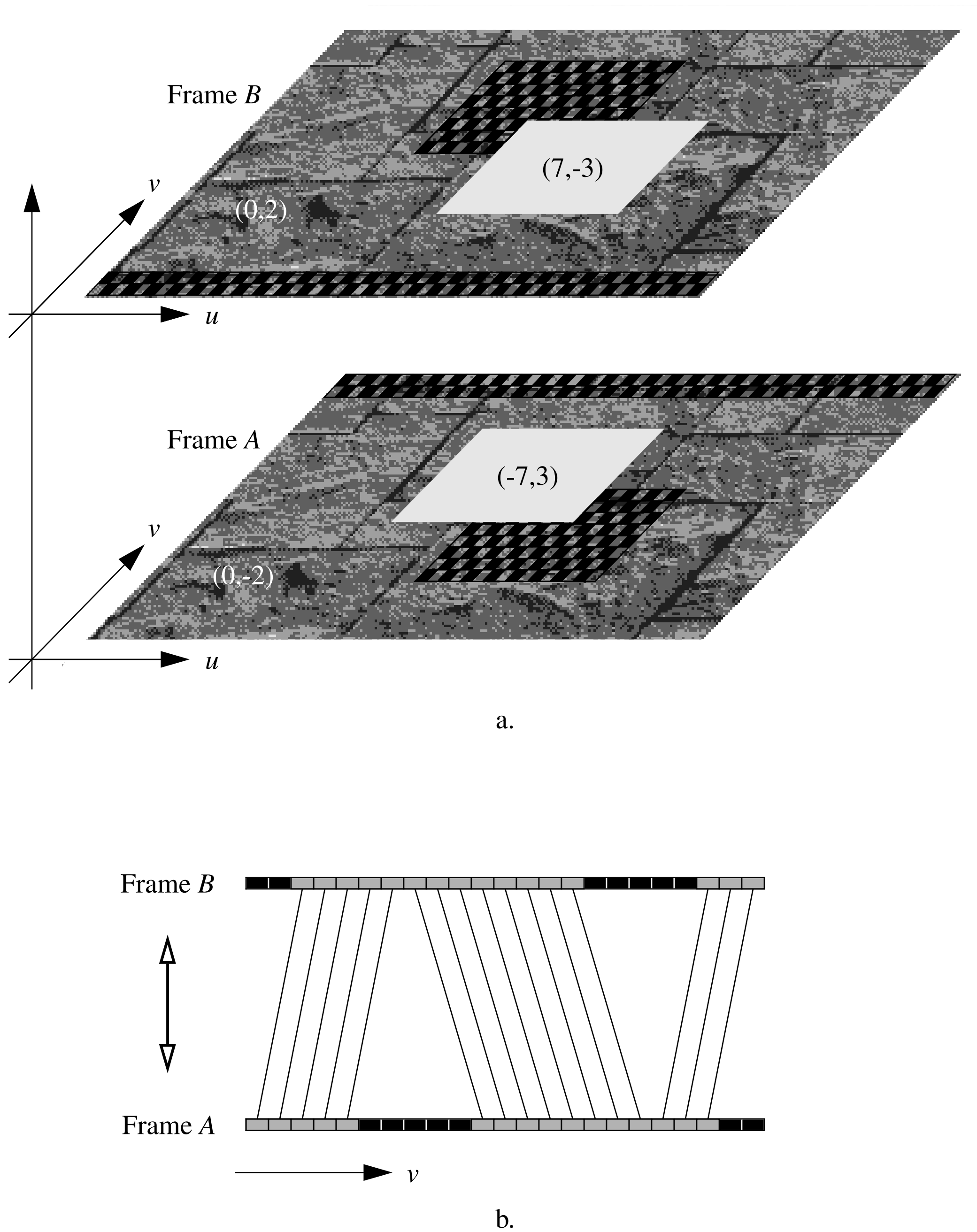


Figure 3.2: Relationship between occlusion and unoccluded region displacement discontinuity, a) sample frames with constant displacement for background and foreground object, b) vertical component of actual displacement along middle image column.

The actual vertical displacement component of the middle image column is shown in Fig. 3.2b. Pixels are indicated by the shaded rectangles along each image column, where occlusions are denoted by the blackened pixels. The slanted-lines connecting unoccluded pixels represent $d_v(29, v)$. We observe that occlusions occur in Frame *B* only at the location of displacement discontinuities in Frame *A*, and vice versa. In particular, for this image column, the number of occluded pixels is equal to the absolute value of the sum of all negative Δd_v's.

The calculation of the exact amount of occlusion becomes more difficult than the simple expression of Eq. (3.14) when dealing with more complexly-shaped objects and situations of multiple, mutually-occluded objects; however, the fundamental relationship between occlusion and displacement discontinuity still holds. Using this property, the location and size of occlusions within the reference frame can be calculated from the discontinuites in the displacement vector field of the current frame.

3.2.3 Reliability of Block-based Estimates

Since we have assumed rigid-body, translational displacement, only a single displacement vector is required to describe the displacement of a contiguous region of pixels. Hence, a block-based technique can be used to estimate these displacement vectors. In these techniques, the current frame is partitioned into a set of non-overlapping blocks, where the blocks may be either fixed- or variable-sized.[3] A search is performed for each block in the current frame for the block in the reference frame that yields the minimum distortion. The estimated displacement vector is merely the vector difference between the location of the original block and the minimum-distortion reference block. Common causes of displacement estimation errors are when: the block has relatively uniform intensity, a displacement discontinuity occurs within a block, or the block contains occluded pixels.

A simple measure of the reliability of an estimate is provided by the prediction signal-to-noise ratio of the respective block, where the noise is defined as the distortion between the original block and the corresponding block in the reference frame. The SNR is a more accurate reliability measure than merely the distortion, due to the problems encountered when the original block has relatively low intensity variation. For these regions, the prediction distortion for all possible

3. The optimal block-dimensions for fixed-sized block-based techniques can be related to the relative cost incurred when coding a block with one or more displacement discontinuities and the density parameter of the generalized Poisson process that models the displacement along each image coordinate (see footnote 2).

displacements within a uniform region in the reference frame will be approximately zero. It is unlikely that the actual displacement for this block will be estimated correctly since multiple reference blocks yield nearly identical distortion. However, the prediction SNR for this block will be low, indicating a high probability of error.

Therefore, the prediction SNR can be used as a rough detection criteria for erroneous estimates. It can also be used to rank displacement estimates, based on the probability of error, when a single displacement vector must be selected from two or more possible estimates.

Chapter 4

Optimal Reference Frame Selection

In this chapter, we examine the problem of selecting the optimum reference frame[1] for the predictive coding of multi-view video signals. We begin with a brief review of the effect of occlusions on prediction performance, and demonstrate the need for adaptive reference frame selection. We then present our novel selection algorithm that estimates the prediction performance that would be obtained from using a particular reference frame without requiring the computationally costly process of displacement estimation for each candidate reference. This approach relies on the relationship between occlusion and displacement discontinuity of unoccluded regions (Section 3.2.2) to select the reference frame that yields the best rate-distortion prediction performance.

The main contributions of this work are: 1) a set of simple tools to eliminate likely false estimates from a noisy displacement vector field, 2) the indirect estimation of a candidate reference frame's displacement field through the combination of multiple, *single-step* DVF's into a *composite* field, and 3) a variance-based measure to estimate the amount of occlusion between two frames from a composite displacement field. This technique achieves impressive prediction and compression performance, while requiring an inconsequential number of bits to specify the selected reference frame and acceptable complexity and storage costs. We illustrate the superiority of this algorithm through comparisons with both non-adaptive and exhaustive search approaches.

4.1 Prediction Performance

Frame-based predictive coders attempt to minimize both the distortion between the predicted and original frames and the number of bits needed to describe the prediction. The rate-distortion performance of the prediction process is related to numerous factors, including the frame sampling

1. The reference frame is said to be optimum when the rate-distortion prediction cost is minimized.

frequency, scene and camera motion, and the sophistication of the predictor's motion model. Probably the most basic factor affecting performance is the amount of occlusion between the reference and desired frames. Since frame-based predictors typically do not distinguish between occluded and unoccluded regions, only regions within the desired frame that are present in the reference frame can be predicted accurately. Conversely, the prediction process will fail for regions within the desired frame that are occluded from the reference frame. Two frames are void of occlusion when the scene objects are static with respect to the viewpoints of each frame; perfect prediction is possible for this situation. The other extreme occurs when two frames have no corresponding points.

Equivalently, prediction performance is related to the *similarity* of the desired and reference frames, which we quantify as the complement of occlusion. Here, our definition of occlusion incorporates all image regions that prove difficult for the particular prediction process used, regardless of the source of this difficulty. Therefore, it is possible for a region to be "visible" to an observer, yet be classified as occluded. For example, since we assume a translational displacement model, if a region undergoes deformation (e.g., morphing) between two frames, an accurate prediction of this region will not be possible and we characterize it as being occluded.

To achieve superior prediction and, hence, compression performance, the reference frame that has minimum occlusion (maximum similarity) with the desired frame should be used in the prediction process. For the case of single-view video signals, the relative amount of occlusion between frames is obvious; in the absence of any inherent camera or scene object motion periodicity, the temporally closest frames generally can be assumed to have the least amount of occlusion with the desired frame. Indeed, this relationship is utilized in the MPEG and H.26x classes of video compression standards, where the temporally closest previous and/or future decoded reference frames are used for the prediction of *P*-, *B*-, and *PB*-frame types.

The similarity relationship between frames in a multi-view video signal is not as straightforward. We next describe three scenarios that illustrate the need for adaptive reference frame selection to ensure that the most similar reference frame is used in the prediction process.

4.2 Multi-view Scenarios

For the first scenario, we consider a video signal consisting of two views with relatively large differential viewpoint parameters, resulting in a high degree of inter-view occlusion. Assuming that the camera and scene object motion is slow with respect to the frame rate, temporally adjacent frames within each view will be quite similar. For this situation, the optimum reference frame would be the frame previous in time and within the same view (i.e., the temporally-offset, intra-view reference frame).

In the second scenario, the multi-view signal is comprised of three views with synchronized temporal sampling and arbitrary viewpoints, where the relative viewpoint parameters are such that a high degree of scene-overlap exists among the views. If the camera and/or scene object motion is rapid or complex, with respect to the predictor's displacement model, the temporally-offset, intra-view frame will have a large amount of occlusion with the desired frame. Instead, a superior prediction will be obtained by using a reference frame from one of the other views, whose temporal index matches that of the desired frame (i.e., a temporally-equivalent, inter-view reference frame). The selection between which of the two possible views to use for the reference frame depends on the camera configuration and the direction of the motion.

Our final scenario demonstrates the possibility of the relative location of the best reference frame being offset from the desired frame in both time and viewpoint. Let the multi-view signal consist of two views, whose relative camera viewpoints are fixed. If the scene is slowly-varying and the absolute viewpoint parameters exhibit motion in the direction of the differential values, a temporally-offset, inter-view frame may yield the best prediction.

Such a scenario is illustrated in Fig. 4.1, where the scene is static, the viewpoints are separated by a horizontal distance of Δc_x for the duration of the sequence, and the camera motion is purely horizontal. The curved, dotted-line traces the motion of the cameras through time, and the thumbnail images depict the spatiotemporal frame locations within each view. Since the scene objects are motionless, the amount of occlusion is related directly to the spatial distance between frames.

Consider the prediction and compression of frames in View 0. Frame $F(1, 0)$ is spatially-closest to $F(0, 0)$ and should be used for the prediction of this frame. For the next two temporal samples, the most similar reference frames are the previous frames in time; i.e., predict $F(0, n)$ from $F(0, n-1)$. From $t = 3T$ till $t = 5T$, the camera velocity is constant, and is equal to $\frac{\Delta c_x}{T}$. The perfect camera position alignment for this scenario yields:

$$F(0, 3) = F(1, 0),\ F(0, 4) = F(1, 3) \text{ and } F(0, 5) = F(1, 4) \tag{4.1}$$

Each of these frames can be reconstructed exactly from knowledge of the appropriate temporal offsets and the corresponding frames in View 1; thus, they can be compressed extremely compactly. The camera motion slows for the final two temporal samples. $F(0, 6)$ is equidistant from

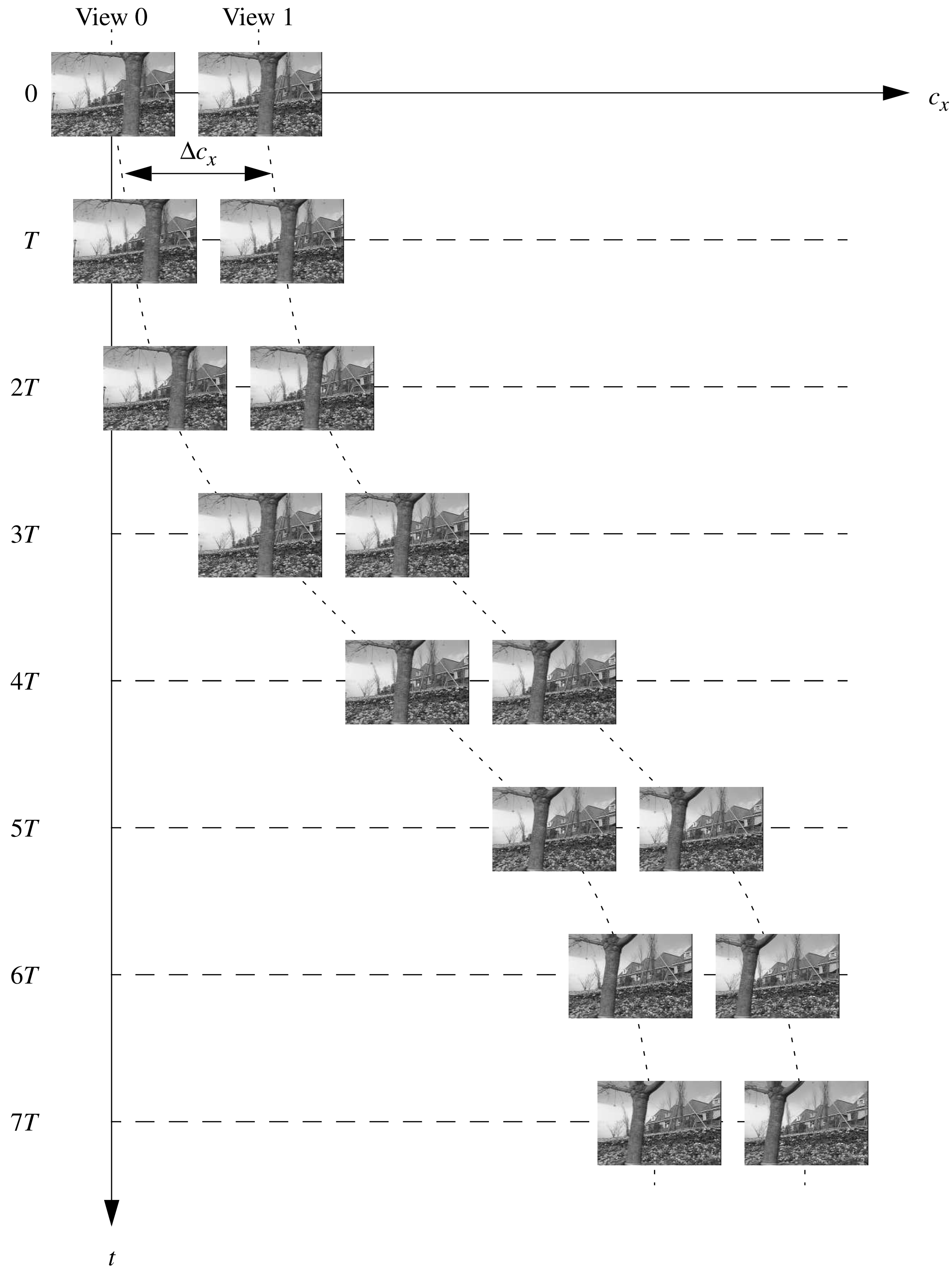


Figure 4.1: Two-view scenario illustrating time-varying location of the optimal reference frame.

Original	F(0,0)	F(0,1)	F(0,2)	F(0,3)	F(0,4)	F(0,5)	F(0,6)	F(0,7)
Reference	F(1,0)	F(0,0)	F(0,1)	F(1,0)	F(1,3)	F(1,4)	F(1,5)	F(0,6)

Table 4.1: Optimal reference frames for View 0 in simulated multi-view signal depicted in Fig. 4.1.

both $F(0,5)$ and $F(1,5)$ – optimum prediction performance would be obtained from using either frame as the reference (or possibly both). The remaining frame should use the previous, intra-view frame. Table 4.1 summarizes the best reference frames for this simulated signal.

From these scenarios, it is evident that the relative location of the optimum reference frame is time-varying and signal-dependent. However, in all prior work known to the author on the predictive coding of multi-view signals, the reference frames have been fixed and heuristically chosen [5, 7, 15, 78, 86, 87]. While the reasoning behind pre-selecting the reference frame has not been explicitly mentioned in the literature, we feel that this approach is due to the naive replication of single-view techniques or the overreliance on the epipolar-line-constraint.

4.3 Early Thoughts on Selection Algorithms

We next describe two approaches for the adaptive selection of the best possible reference frame for the predictive coding of multi-view video signals. The first approach is a brute-force solution that performs an exhaustive search on all candidate reference frames. The second approach utilizes the established relationship between occlusion and prediction performance to perform the selection, and is the foundation of our solution. The relative performance and costs of these techniques are provided in Sections 4.5 and 4.6.

4.3.1 Exhaustive Prediction

Theoretically, the simplest reference frame selection technique would be to predict the desired frame from all possible reference frames and then use the frame that yielded the best prediction. While this method is guaranteed to achieve optimum prediction performance, its implementation is impractical. In many video coders, the prediction operation consumes the largest share of the processing cycles available. The number of prediction operations required at each temporal index for an exhaustive search is quadratic in the number of views.

The complexity of this approach can be reduced through the use of a subband decomposition of the image frames. Here, the image data is consecutively low-pass filtered and sub-sampled by a factor of two in both image directions, resulting in a pyramidal image frame structure and diminished prediction complexity. Depending on the number of pyramid levels, performing the prediction operation on each possible reference frame might be practical. However, due to the loss of higher-frequency image components, the prediction performance for the low-resolution image subband may not be consistent with the performance for the entire frame, and the selection of the optimum reference frame can no longer be guaranteed.

4.3.2 Occlusion-based Selection

An alternative approach for the selection of the optimum reference frame would be to use an estimate of the amount of occlusion as the selection criteria. Previously reported techniques for the detection of occluded regions typically use a measure of symmetry between estimated, bidirectional displacement vectors fields as the decision threshold [13, 83]. The problem with using these occlusion estimation techniques for our reference frame selection task is that the estimation of a displacement vector field is equivalent to performing a frame-based prediction; i.e., we have merely reformulated the exhaustive search method.

We thus redefine the problem of selecting the best reference frame as: develop an estimate of the relative amount of occlusion that avoids performing frame-based prediction for each candidate reference frame. We can simplify this problem by realizing that we only are concerned with the relative amount of occlusion, not the actual location of the occluded regions. Occlusion localization is the goal of conventional occlusion detection schemes, and also the major source of difficulty, due to the circular nature of this problem. We describe our occlusion estimation algorithm that satisfies this complexity constraint in the following section.

4.4 New Algorithm

We calculate the relative amount of occlusion between frames from the degree of discontinuity in the estimated displacement vector field for unoccluded regions, where the requisite vector fields for each candidate reference frame are estimated indirectly.

We first perform a frame-based prediction of the desired frame from only one reference frame. The reference frame for this *single-step prediction* is specified to ensure that any possible reference frame can be reached by stepping along multiple predictions. Since we only want to calculate the degree of discontinuity from reliable, unoccluded region displacement vectors, we make use of broad assumptions to eliminate probable estimation errors. We disregard the single-step displacement estimate of a region if its prediction signal-to-noise ratio falls below a given threshold. We also eliminate displacement estimation errors through the process of vector field reversal, which utilizes the one-to-one relationship of the actual displacement of corresponding points between two images. We complete the unoccluded region detection by discarding displacement estimates along the image border based on the mean displacement vector.

A *composite* displacement vector field for each candidate reference frame then is obtained through simple vector addition of processed single-step fields. For simplicity, we use a slightly modified calculation of the variance of the composite displacement vector field as the measure of discontinuity. The frame with minimum composite vector field variance is assumed to have the least amount of occlusion with the desired frame and is selected as the best reference frame. This approach also can be used to select multiple reference frames – the desired frame is predicted once from each reference frame in the set of N_{ref} frames with the lowest composite field variance; a final prediction then is obtained by combining these predictions.

If the desired frame is to be predicted from only one reference frame, this algorithm requires a maximum of two frame-based predictions: if the minimum variance frame was used in the single-step prediction process, the predicted image is generated from the previously estimated, unprocessed displacement vector field; otherwise, an additional prediction is performed using the selected frame. The major cost of this technique is the large amount of storage needed for the possible reference frames and the retention of the single-step displacement vector fields.

It should be noted that this algorithm can use sub-sampled image data to perform the reference frame selection, as described in Section 4.3.1. Again, a trade-off exists between using higher-resolution image subbands to improve the accuracy of the selection and using lower-resolution data to reduce the complexity. However, since the complexity of the individual steps of eliminating likely estimation errors, generating the composite fields and calculating the displacement variance are on the order of one operation-per-pixel, our approach should be substantially less complex than the sub-band exhaustive search technique.

A flow-chart depicting our reference frame selection algorithm is shown in Fig. 4.2. The various steps of our algorithm are described in detail in the following sub-sections.

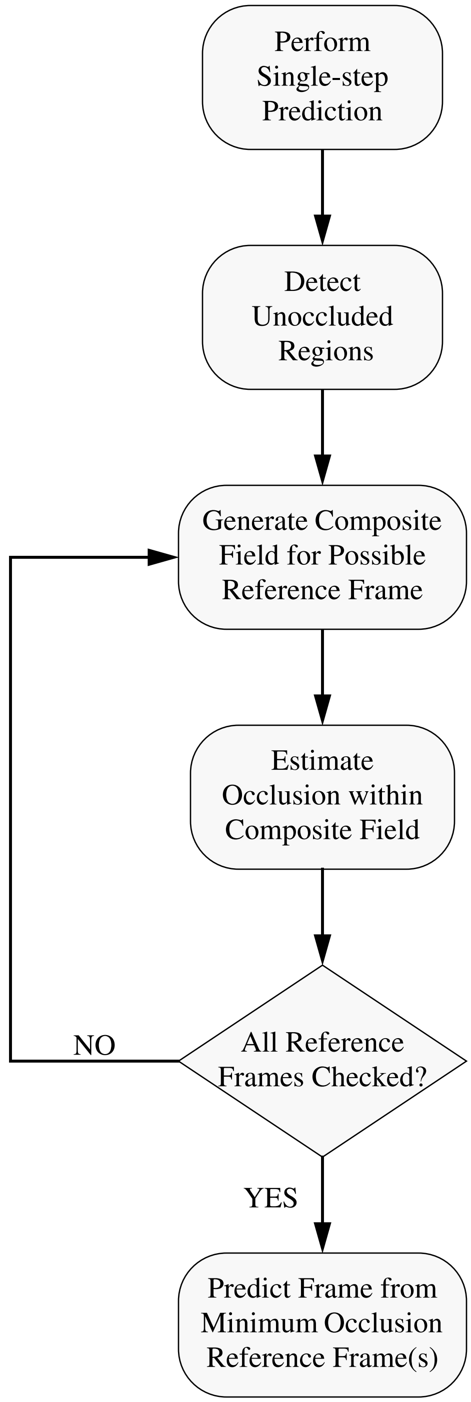


Figure 4.2: Flow-chart for adaptive reference frame selection algorithm.

4.4.1 Composite Displacement Vector Fields

The composite fields for each candidate reference frame are obtained through the vector addition of multiple, processed, single-step displacement vector fields. The single-step fields provide the displacement relationship between two frames in the multi-view signal, and are the basic building block for the generation of the composite fields. Prior to vector addition, the single-step fields are processed to eliminate false estimates that would otherwise corrupt the occlusion measure.

4.4.1.1 Single-step Predictions

The first task in the generation of the composite displacement vector fields is the specification of a set of single-step predictions. These predictions must be ordered to ensure that each possible reference frame can be "reached" through the combination of successive single-step displacement vector fields.

We assign one of the views in the multi-view signal as the *independent* view. This view is coded using any combination of user-specified *I*-, *P*- and *B*-frame types, where the predictively-coded frames use only intra-view reference frames. This stipulation of an independently coded view does not result in an overall loss of compression performance since prediction cannot be performed circularly, e.g., it is impossible to predict both $F(m_1, n_1)$ from $F(m_2, n_2)$ *and* $F(m_2, n_2)$ from $F(m_1, n_1)$. The forward predictions are used as the single-step predictions for the independent view, and the associated displacement vector fields are retained.

A reference view then is assigned for each of the remaining, *dependent* views in the signal. The assignment of the reference views must satisfy the following constraints: 1) at least one dependent view must use the independent view as its reference, and 2) the independent view must be accessible to all other dependent views through successive dependent/reference view links. These constraints ensure a prediction relationship between all views and also avoid any circular predictions. The single-step displacement vector field for each frame in a dependent view is obtained by predicting the desired frame from the frame in its reference view with the same temporal index. All frames within the dependent views will eventually be predicted using an adaptively selected reference frame.

We illustrate the single-step predictions for a three-view signal in Fig. 4.3a, where the bold, vertical bars represent frames within each view. Both Views 1 and 2 reference the independent 0^{th} view, whose frame-type structure, for the five frames shown, is *IBPBP*. The curved arrows denote the single-step predictions of a desired frame (arrow head) from a reference frame (arrow tail).

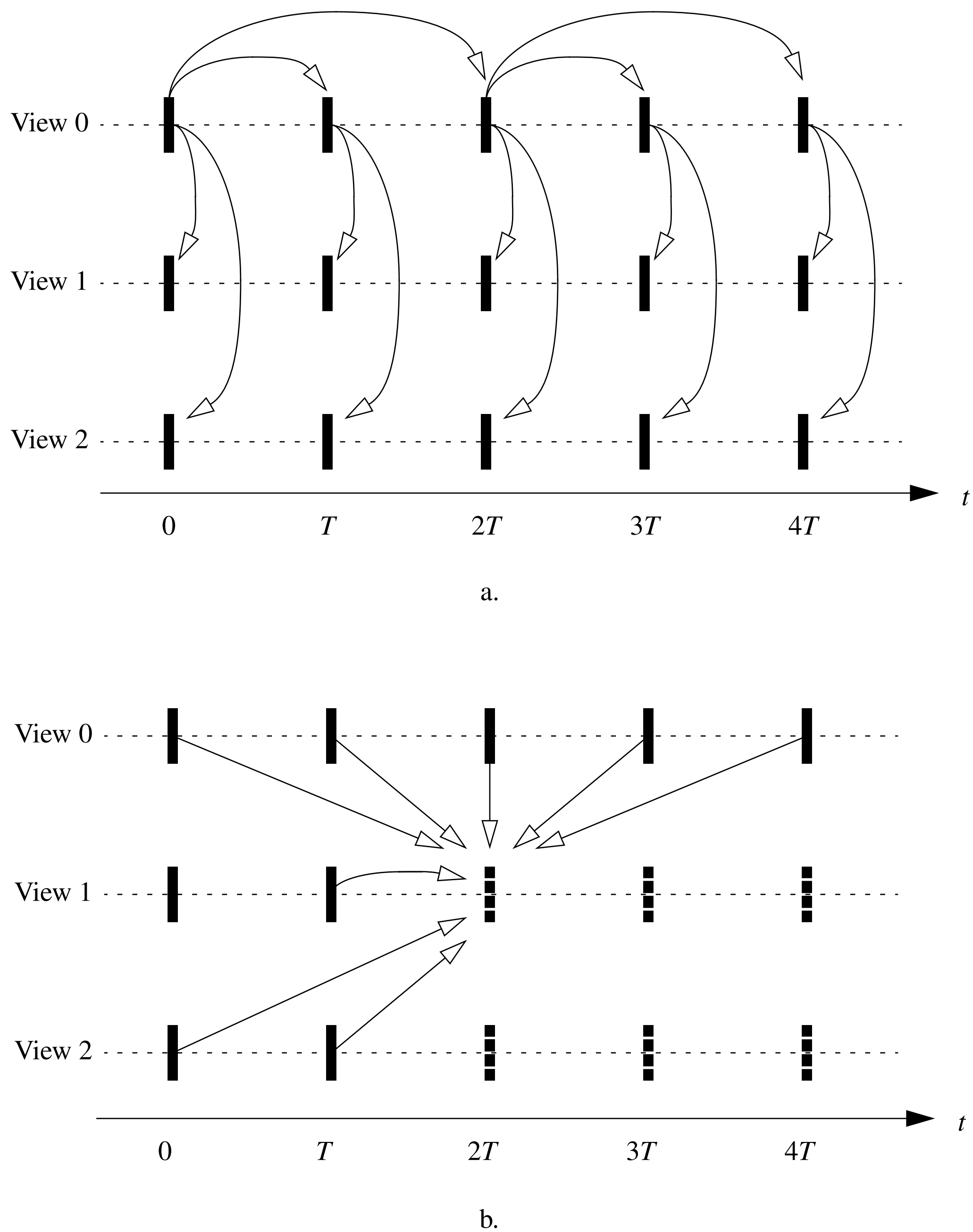


Figure 4.3: Three-view reference frame location scenario, a) single-step predictions, b) possible reference frames for a maximum temporal-offset of two frames.

Only the forward prediction of the bidirectionally-predicted (B) frame is used as a single-step prediction. We observe that any frame within this structure can be reached from any other frame by stepping along successive prediction links. The dependent view predictions are used to step between views, and the independent view's single-step predictions allow for stepping in time.[2] We discuss the actual method of stepping along the single-step predictions to generate the composite displacement vector fields in Section 4.4.1.3.

Since we assume that the minimum-occlusion reference frames will be temporally close to the desired frame, we assign a maximum temporal-index delay, $\Delta n_{\max}$, which specifies the relative temporal range of the possible reference frames. The complete set of reference frames is given by this delay and the independent view's frame type structure. To illustrate the relative reference frame locations, we order the views such that $F(0, n)$ is the independent view, $F(1, n)$ is a dependent view that references the independent view, $F(2, n)$ references either $F(1, n)$ or the independent view, and so on. This ordering was used in Fig. 4.3a. Assuming that only P or B frames are used in the independent view for frames within $n \pm \Delta n_{\max}$, a frame in a dependent view, $F(m \neq 0, n)$, then may be predicted from: 1) any independent view frame within $n \pm \Delta n_{\max}$, 2) the frame previous in time and within the same view, or 3) any other dependent view frame within the negative half of the temporal range, less the same temporal index if the reference view number is greater than that of the desired frame's. The relative locations of the possible reference frames, $F(m_{\mathrm{ref}}, n_{\mathrm{ref}})$, are summarized by,

$$n_{\mathrm{ref}} \in \begin{cases} \{n - \Delta n_{\mathrm{ref}}, \ldots, n + \Delta n_{\mathrm{ref}}\}, & \text{if} \quad m_{\mathrm{ref}} = 0 \\ \{n - \Delta n_{\mathrm{ref}}, \ldots, n - 1\}, & \text{if} \quad m_{\mathrm{ref}} < m \\ \{n - 1\}, & \text{if} \quad m_{\mathrm{ref}} = m \\ \{n - \Delta n_{\mathrm{ref}}, \ldots, n\}, & \text{if} \quad m_{\mathrm{ref}} > m \end{cases} \tag{4.2}$$

Figure 4.3b depicts the possible references frames for the prediction of a desired frame in our three-view signal, where $\Delta n_{\max} = 2$ and the dashed vertical bars represent uncoded frames.

2. If an I-frame type is present in the independent view, frames that have temporal indices equal to or beyond this frame (in the sense of the temporal direction of the stepping) cannot be reached.

Here, a total of eight composite displacement vector fields must be generated and analyzed for the prediction of $F(1, 2)$. For the general case, the total number of possible reference frames for each desired frame, in the m^{th} view, is:

$$\eta_{\text{ref}}(m) = M \cdot \Delta n_{\max} + m + 1 \tag{4.3}$$

4.4.1.2 Unoccluded Region Detection

Since we only want to use the actual displacement vectors for unoccluded regions in our occlusion measure, we wish to exclude any single-step displacement vector estimates that have a high probability of error. Erroneous estimates may occur due to limitations of the prediction process or the meaningless calculation of displacement for occluded regions.

We use the prediction block-SNR as a confidence measure of the accuracy of each displacement estimate, as described in Section 3.2.3. The single-step displacement vector estimate associated with a region whose prediction PSNR falls below a user-defined threshold is assumed to be erroneous, and is discarded from any further consideration.

The thresholding process can be viewed in terms of a classical, binary detection problem [96]. We observe the prediction PSNR for a particular displacement estimate, where the statistics of the PSNR are governed by whether the estimate is either correct or incorrect. The threshold then is based on some knowledge of these statistics and the relative costs of false alarms (i.e, say estimate correct when it is not) and misses. For this application, we are most concerned with obtaining a low probability of false alarm, so we generally will use a conservative (i.e., relatively high) threshold.

We illustrate the displacement vector thresholding in Fig. 4.4 for a one-dimensional case. Figure 4.4a depicts the displacement predicting pixels in Frame *B* from Frame *A,* where a single displacement value was estimated for each set of five neighboring pixels. The prediction signal-to-noise ratios for each 5-pixel region are labelled such that $\text{SNR}_i > \text{SNR}_{i+1}$. Only SNR_5 is less than the given threshold and the associated displacement estimates are eliminated in Fig. 4.4b. Here, the blackened pixels do not necessarily represent occlusions, but merely that the displacement estimate for this region was likely inaccurate.

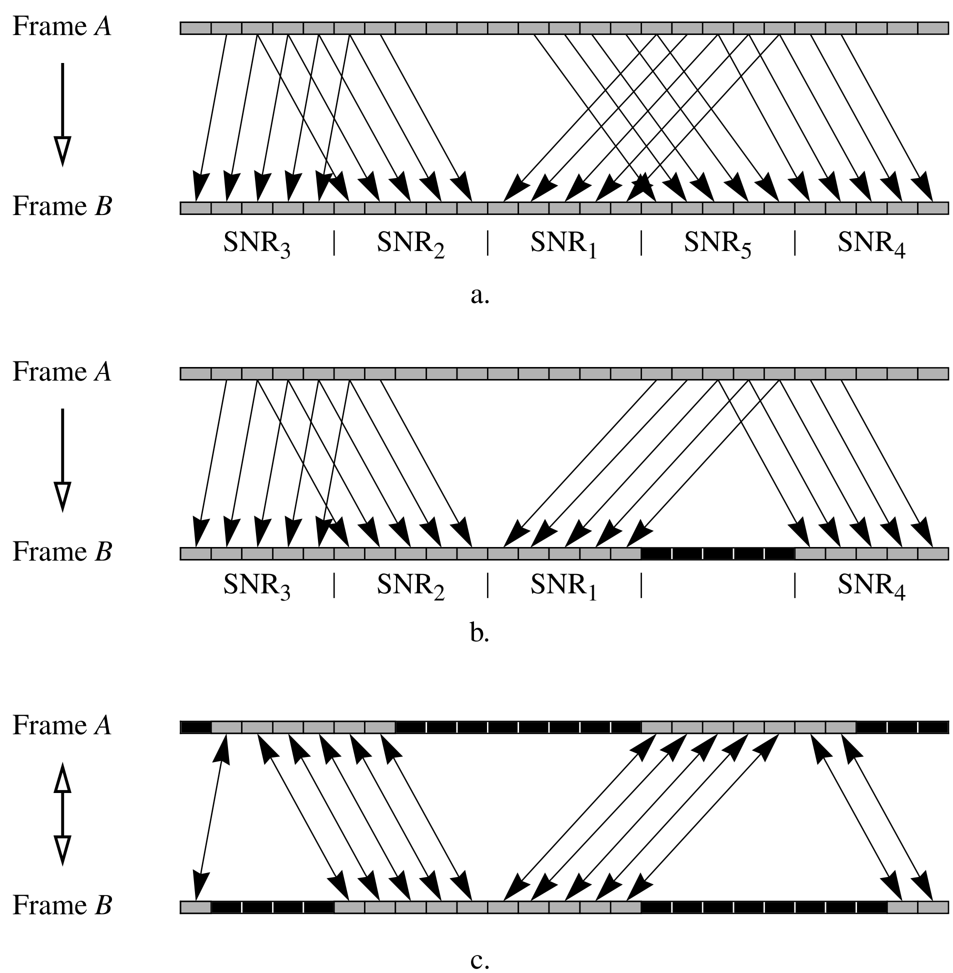


Figure 4.4: Displacement vector thresholding and field reversal, a) original field, b) field after SNR thresholding, c) reversed field.

Once thresholded, we can further eliminate false estimates by using the one-to-one relationship between the true displacement of corresponding points from Frame A to Frame B, and vice versa, satisfies:[3]

$$\boldsymbol{d}_{A \leftarrow B}(u - d_{u_{B \leftarrow A}}(u, v), v - d_{v_{B \leftarrow A}}(u, v)) = -\boldsymbol{d}_{B \leftarrow A}(u, v) \tag{4.4}$$

If the estimated displacement vector field is inconsistent with this relationship, we assume that an error has occurred and eliminate the likely false estimate.

3. See Eq. (3.9) in Section 3.2.1.

Given a single-step displacement vector field in one direction, we examine the usage of the pixels in the reference frame. If a reference pixel is used for the prediction of only one pixel in the desired frame, we assume that this displacement estimate is accurate and assign the reversed displacement vector according to Eq. (4.4). Conversely, if a pixel is referenced by more than one displacement vector, we are assured that all of these estimates cannot be correct. Again, we use the prediction SNR for the decision as to which of these estimates has the lowest probability of error. The displacement estimate for the reference pixel is assigned the negative of the forward displacement vector with maximum prediction SNR, and all other vectors are invalidated. Finally, if a pixel in the reference frame is not used in the prediction, we assume that this pixel is occluded and do not assign a reversed displacement estimate to it. The field reversal process not only detects unoccluded regions, but also provides valid displacement vector estimates in both directions that satisfy Eq. (4.4), i.e., it generates $\hat{\boldsymbol{d}}_{A \leftarrow B}(u, v)$ from $\hat{\boldsymbol{d}}_{B \leftarrow A}(u, v)$. This information will be used to step along single-step predictions in both directions.

Figures 4.4b and 4.4c illustrate the process of eliminating displacement estimates that do not satisfy the one-to-one mapping of actual unoccluded region displacements. The bidirectional arrows in Fig. 4.4c indicate that these displacement values map corresponding pixels from Frame *A* to Frame *B*, and vice versa. Again, the blackened pixels represent regions whose displacement estimate has been invalidated. We note that even though the displacement was estimated for contiguous regions of five pixels, only individual pixel displacement vectors are eliminated. Since the displacement estimate might be accurate for portions of the region, we do not eliminate the displacement for entire regions if a subset of its displacements is inconsistent with Eq. (4.4).

The final step in the detection of highly reliable displacement estimates is the elimination of displacement vectors along the image borders. We assume that the global displacement of the image background between image frames is given by the mean displacement vector of the single-step field after thresholding and field reversal. If the mean vector is non-zero, pixels along the appropriate horizontal and vertical image borders are likely occluded. The width and height of the occluded region are, respectively, equal to the magnitude of the horizontal and vertical components of the mean vector, and the appropriate border sides depend on the sign of the individual components. For example, if the mean displacement vector is $\begin{bmatrix} -3 & 5 \end{bmatrix}^t$, we assume that a border occlusion exists for the vertical strip of three pixels along the left image border and horizontal strip of five pixels along the top border.

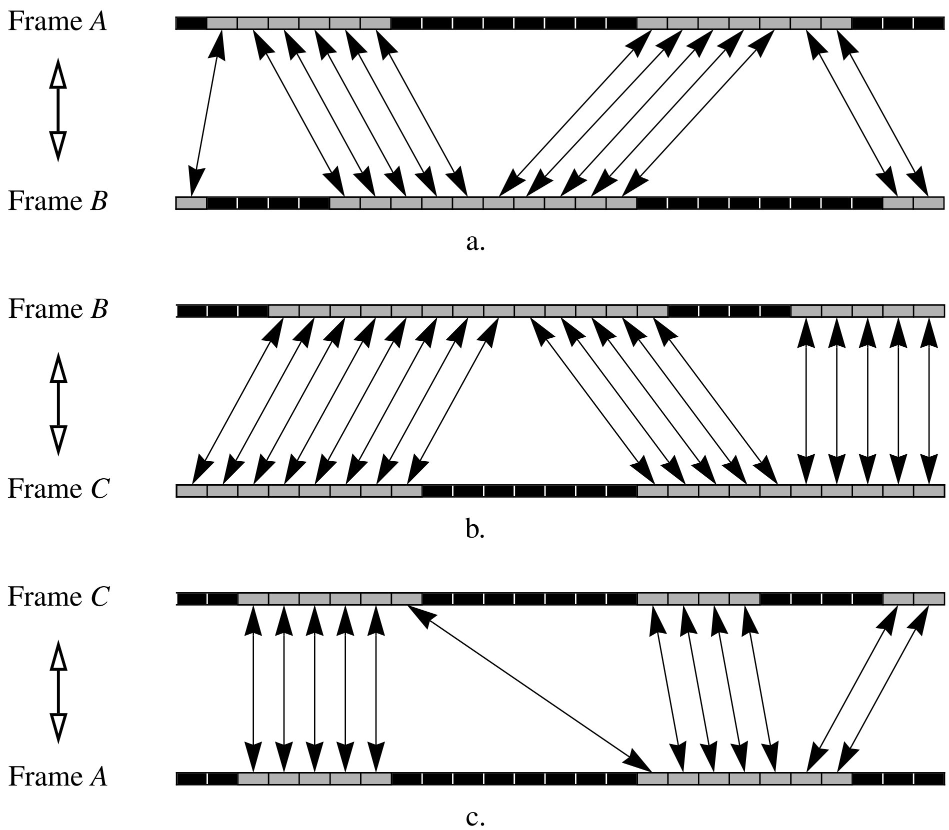


Figure 4.5: Generation of composite displacement vector field through field addition, a) and b) single-step displacement fields, c) composite field.

We first calculate the mean vector for all valid vectors in the single-step field and round the vector to integer values. Here, we invalidate an entire image region of constant displacement if one or more pixels within the region lies within the border occlusion. This differs from our approach to the elimination of displacement estimates that were inconsistent with the one-to-one mapping of actual displacement vectors. Since only a relatively limited number of image regions will lie within the border occlusion, we can afford to be much more cautious as to which estimates to retain.

4.4.1.3 Vector Field Addition

The composite displacement vector field for each possible reference frame is obtained through the simple vector addition of the processed single-step fields. Given two displacement vectors fields predicting Frame A from Frame B ($\hat{\boldsymbol{d}}_{A \leftarrow B}(u, v)$), and Frame B from Frame C ($\hat{\boldsymbol{d}}_{B \leftarrow C}(u, v)$), the composite field relating frames A and C is given by,

$$\hat{\boldsymbol{d}}_{A \leftarrow C}(u, v) = \hat{\boldsymbol{d}}_{A \leftarrow B}(u, v) + \hat{\boldsymbol{d}}_{B \leftarrow C}(u - \hat{d}_{u_{A \leftarrow B}}(u, v), v - \hat{d}_{v_{A \leftarrow B}}(u, v)) \tag{4.5}$$

where the combined fields may be either single-step or composite fields. If Frame C is actually Frame A, substituting Eq. (4.4) into (4.5) yields $\hat{\boldsymbol{d}}_{A \leftarrow A}(u, v) = \boldsymbol{0}$, as expected. The normal associative and commutative properties also hold for the addition of vector fields.

The generation of composite fields relies on the property that a region unoccluded in Frames A and B and also unoccluded in Frames B and C will be unoccluded between Frames A and C.[4] The displacement of unoccluded regions between the desired frame and its possible reference frames, thus, can be estimated without the costly process of displacement estimation for each reference frame. However, it should be noted that the composite displacement vector fields cannot be used directly to provide an accurate estimate of prediction performance. The composite fields are generated for regions that are assumed to be unoccluded, and do not cover the entire image frame. The resulting prediction performance from using these fields will likely be unrelated to the actual performance for the complete frame.

In Fig. 4.5, we illustrate the addition of two displacement fields in one-dimension, where the field relating Frames A and B is the processed single-step field from Fig. 4.4a. The majority of the displacements of the resulting composite field between Frames A and C are close to zero, with only one relatively large estimate in the middle of the field.

4.4.2 Occlusion Measure

For each composite field, we approximate the displacement discontinuity, and hence amount of occlusion, from a modified variance calculation of valid displacement estimates. The modification is the result of augmenting the composite field with a region of zero-displacement vectors. If used in its initial, unaltered form, the variance of the composite DVF provides a measure of the relative amount of inter-object occlusion. By augmenting the DVF, occlusions along the image borders, due to the displacement of the background, are also accurately estimated.

4. The converse of this property obviously does not hold; nor will a region in Frame A that is occluded from Frame B necessarily be occluded from Frame C.

For example, consider a field where each valid displacement estimate equals the same non-zero vector. The variances of the horizontal and vertical components of this field obviously will be zero. However, occlusions are present on the image border, which may represent a substantial portion of the entire frame.

We assume that the background has constant displacement equal to the mean displacement vector of composite field. The mean values, $\mu_{\hat{d}_u}$ and $\mu_{\hat{d}_v}$, are calculated from the valid displacement estimates, and are substituted for the displacement discontinuity components in Eq. (3.14) to yield the size of the augmented region:

$$\text{augmented size} = U \cdot \left|\mu_{\hat{d}_u}\right| + V \cdot \left|\mu_{\hat{d}_v}\right| - \left|\mu_{\hat{d}_u}\right| \cdot \left|\mu_{\hat{d}_v}\right| \tag{4.6}$$

where the $\max(\ )$ functions from Eq. (3.14) have been removed since we are assured that $U \geq \mu_{\hat{d}_u}$ and $V \geq \mu_{\hat{d}_v}$. Finally, the relative occlusion measure between the desired and possible reference frame is given by the sum of the variances of the displacement vector components of the augmented field:[5]

$$\text{occlusion measure} = \sigma^2_{\hat{d}_u} + \sigma^2_{\hat{d}_v} \tag{4.7}$$

This occlusion measure is a function of both the mean and variance of the composite displacement vector field, and, thus, adequately approximates both inter-object and background displacement discontinuity.

Returning to our example one-dimensional field in Fig. 4.5, we observe that the occlusion measure for the field in Fig. 4.5c is significantly less than the top field, and Frame *C* would be selected as the reference frame for the prediction of Frame *A*.

4.5 Cost Analysis

In this section, we compare the relative encoder complexity and storage costs of using the described, adaptive algorithm versus an exhaustive search for the selection of the optimal reference frame; these costs are fixed for the multi-view decoder. Since the prediction and coding of the

5. We have experimented with a more sophisticated occlusion measure based on a direct application of Eq. (3.13), only to find the simple variance calculation more robust to spurious estimates.

independent views are identical for both selection methods, the analysis is performed only on the dependent views. A fixed, pre-selected reference frame scheme is used as the baseline for the comparison. We assume that the desired frame is predicted from N_{ref} reference frames (typically one or two frames).[6]

4.5.1 Complexity

For a fixed reference frame scheme, the total complexity to generate the final predicted image is given by,

$$C_{fixed} = N_{ref} \cdot C_{pred} + C_{comb}(N_{ref}) \tag{4.8}$$

where C_{pred} is the complexity of performing a frame-based prediction and $C_{comb}(N_{ref})$ is the complexity of combining multiple predictions, with $C_{comb}(1) = 0$. If a full-search prediction process is used, C_{pred} is on the order of the search area in pixels, where we have assumed, for simplicity, that the search ranges for each reference frame are equal.

The prediction of the desired frame using an exhaustive search selection technique amounts to: 1) generating a prediction from each possible reference frame, 2) calculating the prediction distortion for each reference, and 3) selecting and combining the N_{ref} minimum distortion predictions. The average number of possible reference frames, for each desired frame, is obtained by averaging Eq. (4.3) over the $(M-1)$ dependent views:

$$\eta_{ref\text{-}ave} = \left(\Delta n_{max} + \frac{1}{2}\right)M + 1 \tag{4.9}$$

which is obviously greater than or equal to N_{ref}. Since the prediction distortion is a by-product of the prediction operation and the reference frame ranking is performed for the entire frame, we dis-

6. If N_{ref} is greater than one, the individual predictions typically are combined into a final prediction through selection and/or averaging of regions within each prediction.

card its calculation from the per-pixel complexity calculation. The total complexity for the exhaustive reference frame selection technique is given by,

$$C_{\text{exhaust}} = \eta_{\text{ref-ave}} \cdot C_{\text{pred}} + C_{\text{comb}}(N_{\text{ref}}) \tag{4.10}$$

The prediction of a desired frame using our adaptive reference frame selection algorithm consists of: 1) performing a single-step prediction, 2) detecting unoccluded regions in the single-step field, 3) generating and calculating the variance of the composite field for each possible reference, and 4) predicting the desired frame from the N_{ref} selected references, less one prediction if the single-step prediction is selected. Letting $Pr\{\text{not selected}\}$ denote the probability that the single-step prediction is not selected for the final prediction, the average complexity of the adaptive reference frame selection algorithm is:

$$C_{\text{adapt}} = (N_{\text{ref}} + Pr\{\text{not selected}\})C_{\text{pred}} + C_{\text{comb}}(N_{\text{ref}}) + 4\eta_{\text{ref-ave}} + 5 \tag{4.11}$$

where the last two quantities are, respectively, due to the generation of the composite fields (2 operations per pixel per reference for both the addition of the vector fields and the calculation of the displacement variance) and the detection of unoccluded regions (one operation per pixel for the thresholding, two for the field reversal, and another two for the calculation of the mean displacement vector).

The complexity overhead of the exhaustive technique versus our algorithm is:

$$C_{\text{exhaust}} - C_{\text{adapt}} = [\eta_{\text{ref-ave}} - (N_{\text{ref}} + Pr\{\text{not selected}\})]C_{\text{pred}} - 4\eta_{\text{ref-ave}} - 5 \tag{4.12}$$

The bracketed term is always greater than or equal to zero, and it equals zero only if the desired frame is to be predicted using all possible reference frames, i.e., $N_{\text{ref}} = \eta_{\text{ref-ave}}$. Assuming that not all possible references are used, the adaptive selection technique is more complex only if:

$$C_{\text{pred}} < \frac{4\eta_{\text{ref-ave}} + 5}{\eta_{\text{ref-ave}} - N_{\text{ref}} - Pr\{\text{not selected}\}} \tag{4.13}$$

For a three view signal, with a maximum temporal delay of one frame and a single reference frame, Eq. (4.13) is satisfied if the prediction of the desired frame from one reference requires, on average, less than 9 operations per pixel. This would restrict a full-search prediction technique to have a maximum search range of only ±1 pixel in both image coordinates.

This complexity analysis has been performed assuming that full-resolution image frames were used in the exhaustive search and single-step predictions. Similar relative complexity costs would be obtained if these predictions were performed using decimated imagery.

4.5.2 Storage

A frame used in the prediction of a desired frame obviously must be stored in memory until the last frame that uses the reference has been coded. To facilitate our rate-distortion comparisons (Section 4.6), we assume that the fixed reference frame scheme may use any of the possible references frames available for the exhaustive search and adaptive techniques.[7] The maximum number of reference frames for the dependent views is,

$$\eta_{\text{ref-max}} = (\Delta n_{\max} + 1)M \tag{4.14}$$

In addition to these reference frames, the independent view requires at most two additional references, and the image frame currently being coded must be reside in some physical memory. This yields a total of $(\eta_{\text{ref-max}} + 3)$ frame stores, and a total storage for both the fixed-reference and exhaustive reference frame selection techniques of:

$$S_{\text{fixed}} = S_{\text{exhaust}} = (\eta_{\text{ref-max}} + 3) \cdot (\text{bytes per frame}) \tag{4.15}$$

The adaptive reference frame selection algorithm must store not only all the frames required for the fixed-reference scheme, but also the processed single-step displacement vector fields (DVF) for each possible reference frame. The maximum storage for our algorithm is given by,

$$S_{\text{adapt}} = S_{\text{fixed}} + \eta_{\text{ref-max}} \cdot (\text{bytes per DVF}) \tag{4.16}$$

Since each pixel within the processed field is described by a flag indicating whether the pixel is valid along with a two-dimensional displacement vector if valid, the adaptive algorithm requires

---

7. Since a practical implementation of a fixed reference scheme will most likely only use the temporally and/or spatially closest reference frames, the storage requirement will be reduced accordingly.

approximately 2-3 times the storage of the exhaustive search and fixed reference frame methods. A reduction in the relative storage cost of our technique can be obtained through the use of subsampled single-step displacement fields.

4.6 Experimental Results

The fixed, exhaustive search, and adaptive reference frame selection algorithms were applied to the multi-view signals described in Section 2.4. All frame predictions were performed using a full-search, block-based technique with 16×16 pixel blocks and half-pixel accuracy. For simplicity, the independent view in each sequence was coded using only forward prediction, similar to the H.261 standard. The search range was specified for each sequence to ensure that it covered the range of the actual displacement of corresponding points between the desired and candidate reference frames.

The displacement vector for each image block was estimated using the mean-absolute-difference (MAD) distortion function. Consequently, the exhaustive reference frame selection technique used the predicted-block MAD, summed over the entire image frame, as its selection criteria. The adaptive selection algorithm also used this distortion for the elimination of likely displacement estimation errors; i.e., an image block was assumed to be unoccluded if its variance-to-MAD ratio was greater than a specified threshold. A constant threshold of 25 was used for all sequences examined.

The predicted frames were coded using a modified MPEG encoder capable of handling multiple views. A fixed DCT coefficient quantization matrix was used, resulting in a constant quality coded sequence.

As an initial test, the adaptive reference frame selection algorithm was applied to a simulated multi-view signal consisting of two sets of frames from the *Flower Garden* sequence, where identically indexed frames in each view were offset by six temporal samples. The artificial motion of the cameras was specified to coincide with that of Fig. 4.1, with the original sequence's temporal indices for the frames in each view given in Table 4.2. View 1 was defined as the independent view, and a maximum temporal delay of three frames was permitted.

The reference frames selected by the adaptive algorithm are provided in Table 4.3. Except for the selection error for the first frame, these are the exact reference frames that were discussed in Section 4.2 (see Table 4.1). The only frame that we had previously speculated might have a different optimal reference frame was $F(0, 6)$. This frame is spatially equidistant from frames $F(0, 5)$,

	Temporal Index							
	0	1	2	3	4	5	6	7
View 0	0	1	3	6	12	18	21	22
View 1	6	7	9	12	18	24	27	28

Table 4.2: *Flower Garden* sequence's original temporal indices for simulated two-view signal.

Original	F(0,0)	F(0,1)	F(0,2)	F(0,3)	F(0,4)	F(0,5)	F(0,6)	F(0,7)
Reference	F(1,1)	F(0,0)	F(0,1)	F(1,0)	F(1,3)	F(1,4)	F(1,5)	F(0,6)

Table 4.3: Adaptively selected reference frames for the original frames in View 1 of the simulated signal.

$F(1, 4)$ and $F(1, 5)$, and the resulting occlusion measures for these references were, respectively, 38.8, 42.5, and 30.8; the other possible references for this frame had occlusion measures over twice these values.

For a more meaningful evaluation, the selection algorithm was applied to numerous, real multi-view signals. The first results that we will examine were obtained from the *Finish Line* sequence. View 0 was coded independently, a maximum temporal delay of one frame was used, and each of the frames in the dependent view was predicted from only one reference frame. The 51st temporally-indexed frame within View 1, $F(1, 50.5)$, is shown in Fig. 4.6, along with its possible references and their relative temporal locations. The composite displacement vector fields for each of these reference frames are shown in Fig. 4.7, where the left- and right-columns, respectively, represent the horizontal and vertical image components of the estimated displacement vectors. Each discernible variation in intensity represents a displacement discontinuity of one pixel. Figure 4.7a illustrates the unprocessed, single-step displacement vector field obtained from the prediction using $F(0, 51)$.The single-step field after unoccluded region detection is shown in Fig. 4.7b, where the invalidated pixels have been blackened. Figures 4.7c and 4.7d show the composite fields for the possible references $F(0, 50)$ and $F(1, 49.5)$, obtained through the field addition of the appropriate, processed single-step fields. Clearly, the valid displacement estimates within the composite field for reference $F(1, 49.5)$ are significantly more uniform than those of the other

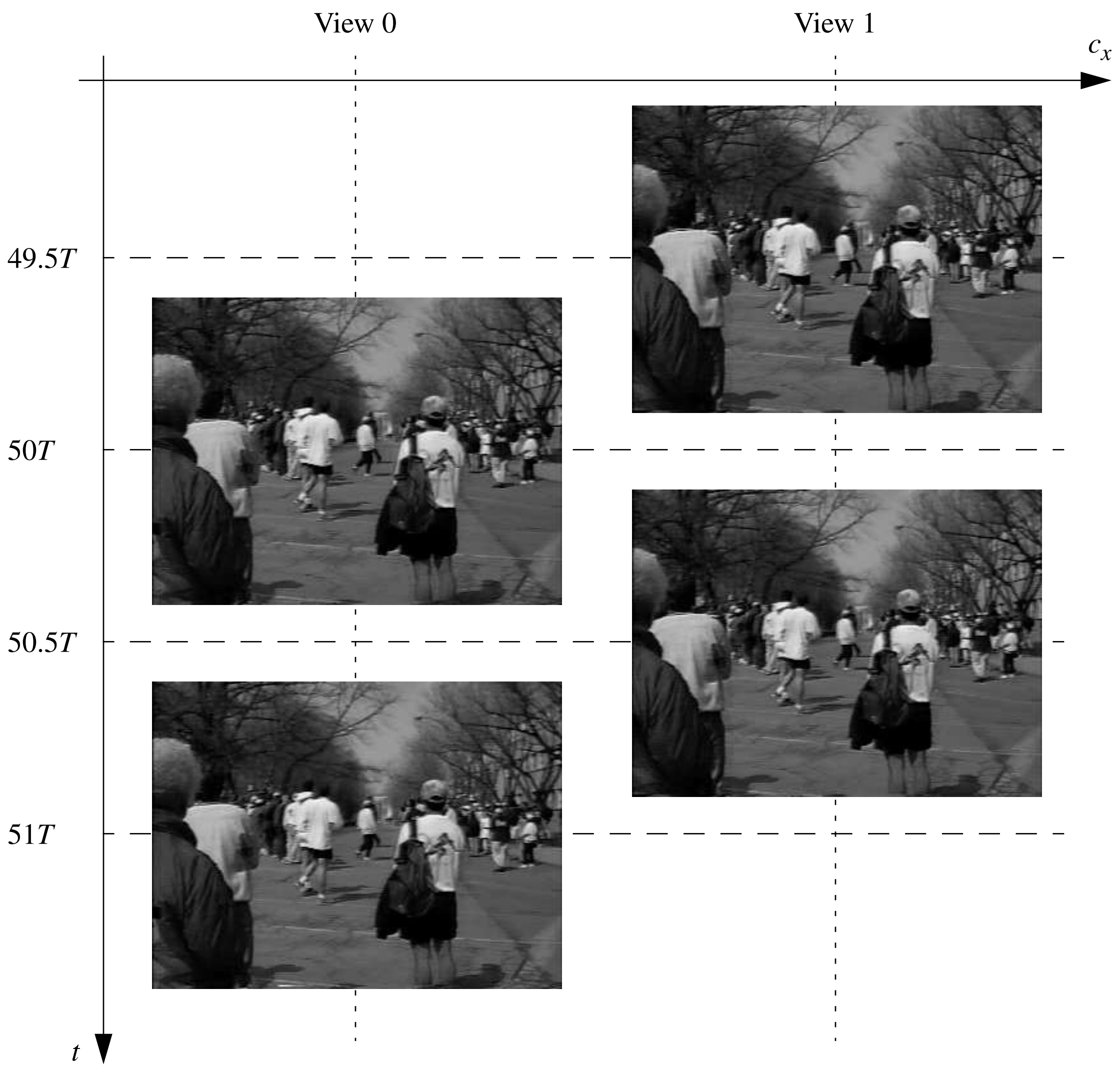


Figure 4.6: Temporal location of possible references for $F(1,50.5)$ in *Finish Line* sequence.

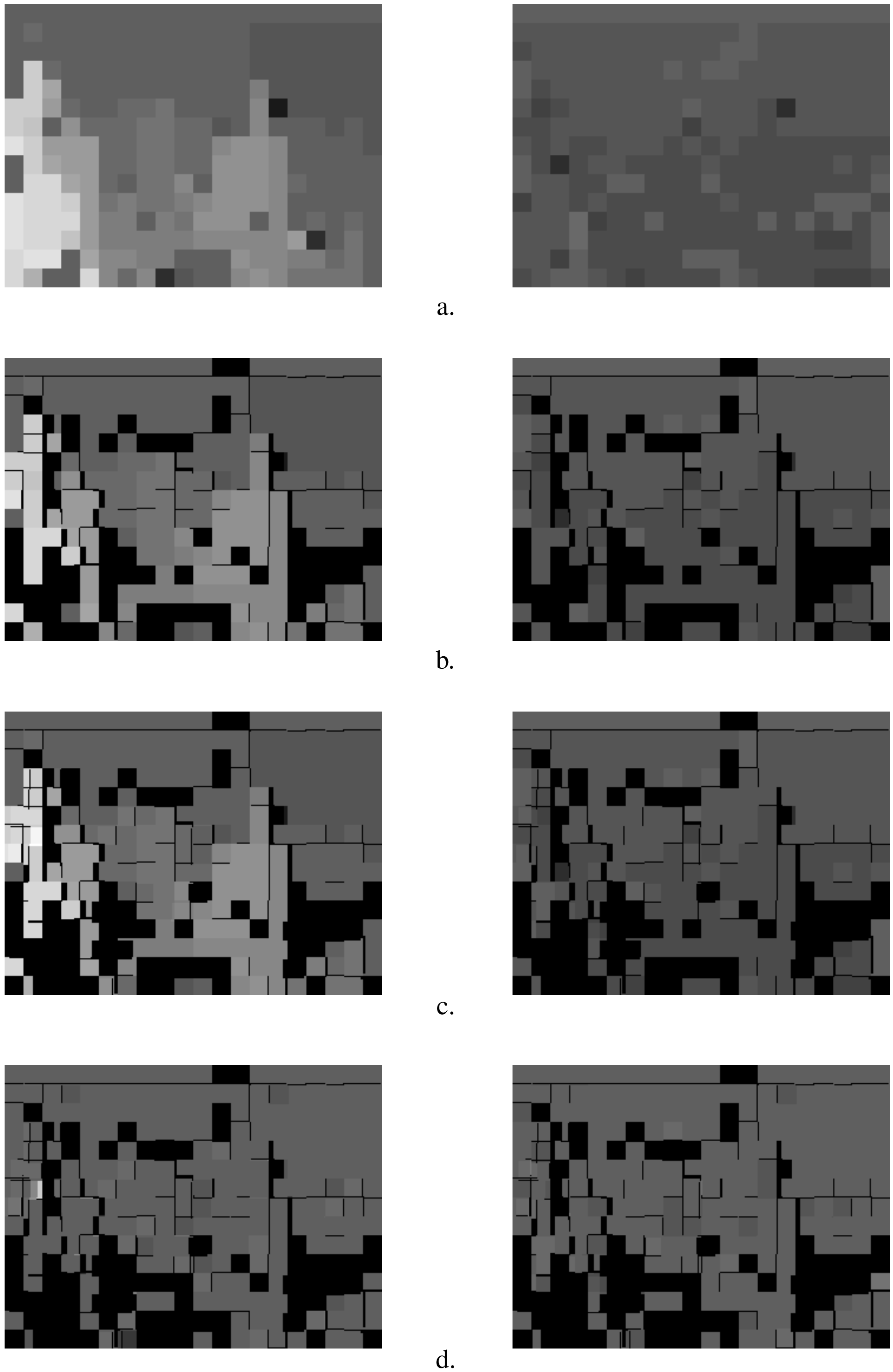
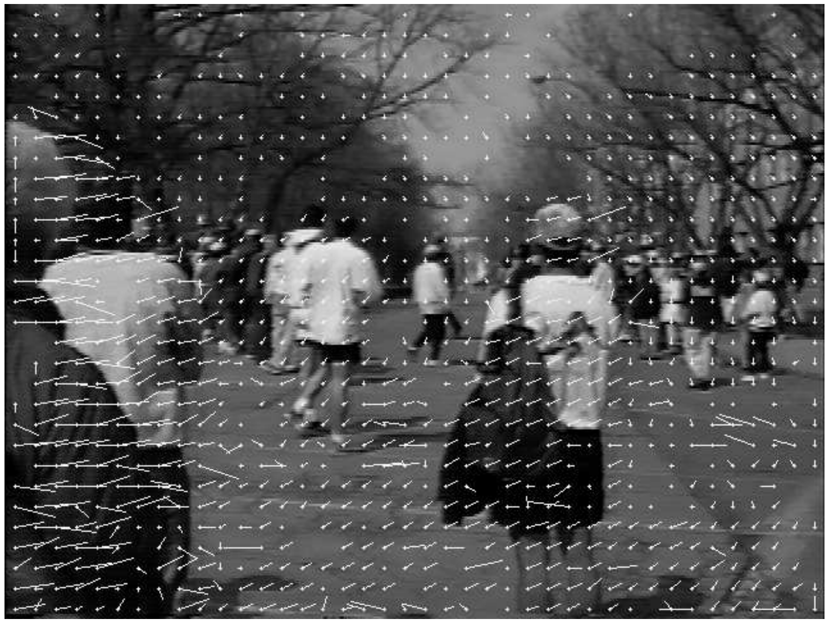


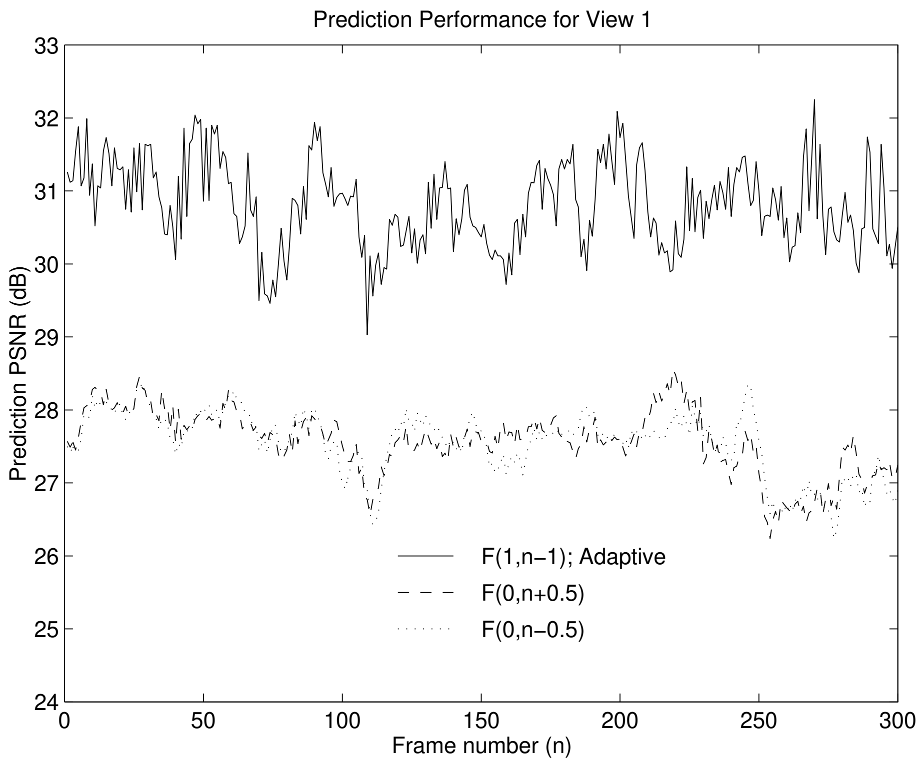
Figure 4.7: Horizontal (left-column) and vertical (right-column) displacement components for *Finish Line* $F(1,50.5)$, a) single-step field from $F(0,51)$, b) processed single-step field, c) composite field for $F(0,50)$, d) composite field for $F(1,49.5)$.



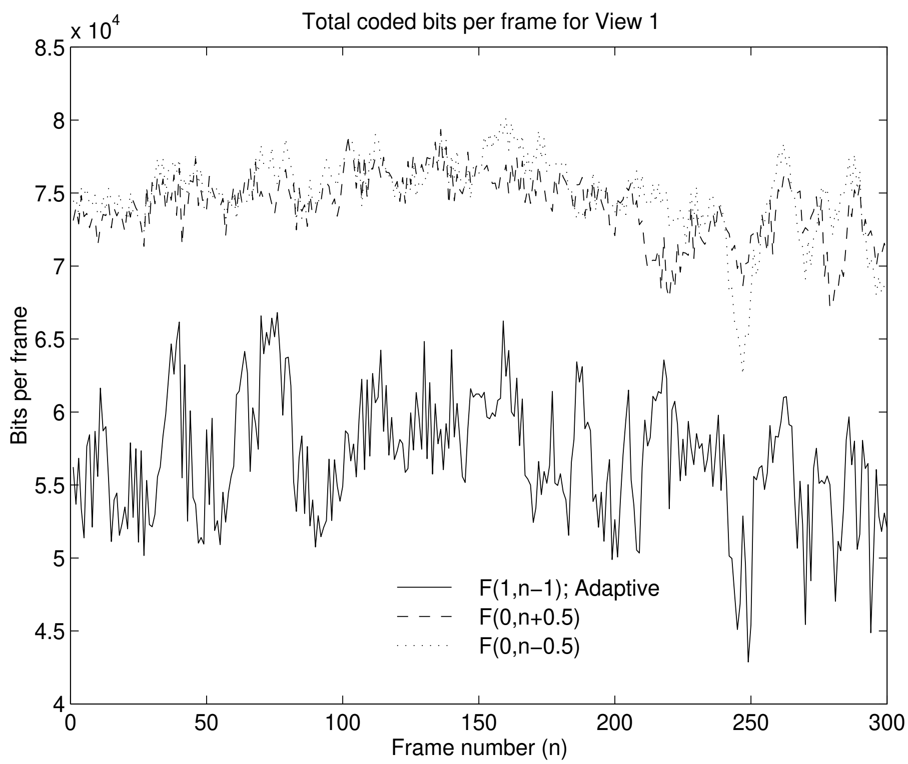
a.

b.

Figure 4.8: Graphical display of estimated displacement vectors for $F(1,50.5)$, a) reference $F(0,51)$, b) reference $F(1,49.5)$.



a.



b.

Figure 4.9: *Finish Line* performance for possible reference frames, a) prediction PSNR, b) total coded bits per frame.

	Reference Frame Ranking		
	Best	2nd Best	Worst
$F(1,n\text{-}1)$	55	126	89
$F(0,n+0.5)$	151	97	22
$F(0,n\text{-}0.5)$	64	47	159

Table 4.4: Reference frame ranking, with respect to prediction PSNR, for the *Skate*r sequence

possible reference frames. The occlusion measures for $F(0, 51)$, $F(0, 50)$ and $F(1, 49.5)$ were 45.9, 49.3 and 7.6, respectively; thus, the intra-view reference was selected for the prediction of this frame.

Figure 4.8 illustrates the ability of the algorithm to generate accurate displacement vector fields for unoccluded regions through the use of composite fields. Here, estimated fields obtained from the frame-based prediction of $F(1, 50.5)$ from reference frames $F(0, 51)$ and $F(1, 49.5)$ are overlaid on the original image. Estimated vectors with non-zero magnitude are represented by the white lines, positioned at the center of the 16×16 pixel blocks. Again, we observe that the field for reference $F(1, 49.5)$ is considerably more uniform than that for $F(0, 51)$.

The per-frame prediction PSNR and bit count for View 1 in the *Finish Line* sequence are shown in Fig. 4.9, where the three possible reference frames for $F(1, n)$ are denoted by $F(0, n+0.5)$, $F(0, n-0.5)$ and $F(1, n-1)$. While the relative location of the optimal reference frame does not vary for this sequence, this location would likely not be known a priori. In fact, if either inter-view reference frame was used, the average prediction PSNR would have decreased by over 4 dB and the coded bit rate would have increased by more than 40% when compared to the intra-view reference, which was correctly selected by our algorithm.

The two-view *Skater* sequence was coded using the same reference frame parameters as the *Finish Line* sequence. The per-frame prediction PSNR over the entire sequence for the fixed and adaptively selected reference frame schemes are shown in Fig. 4.10a. We observe that for the first 40 frames and the last 20 frames the intra-view reference frame, $F(1, n-1)$, yields a prediction gain of approximately 1-1.5 dB over the two inter-view references. From frames 175 to 225, the best reference is one of the two inter-view reference frames. For the remainder of the sequence, all references achieve roughly the same performance. This variation validates our hypothesis that the

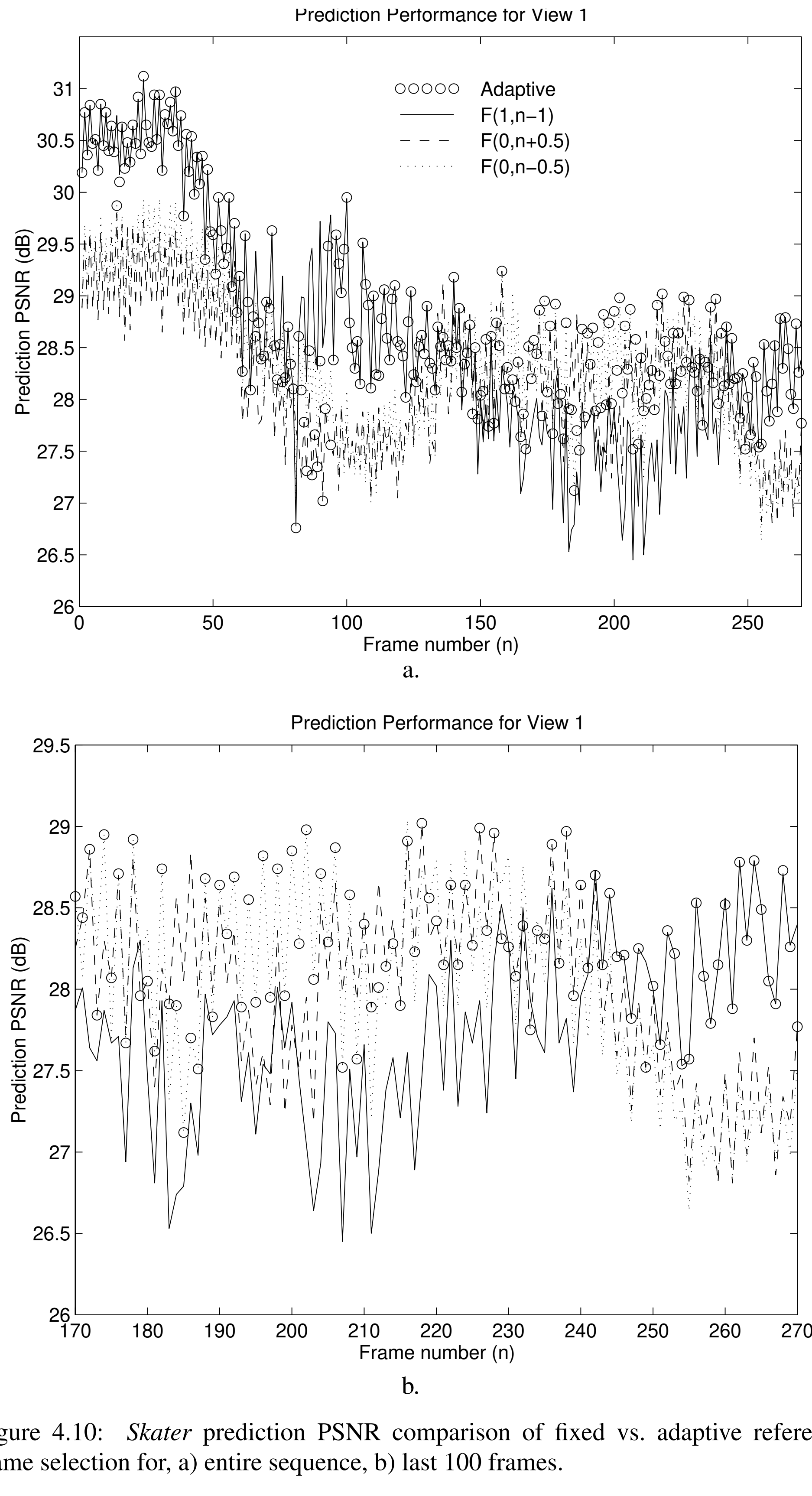


Figure 4.10: *Skater* prediction PSNR comparison of fixed vs. adaptive reference frame selection for, a) entire sequence, b) last 100 frames.

	Toy Train		*Piano*		*Manege*	
	PSNR (db)	Rate (Mbps)	PSNR (dB)	Rate (Mbps)	PSNR (dB)	Rate (Mbps)
Fixed	27.04	2.99	27.62	2.54	23.18	4.08
Exhaustive	28.70	2.32	30.79	1.79	26.09	2.96
Adaptive	28.69	2.32	30.79	1.79	25.92	2.93

Table 4.5: Average prediction PSNR and coded bit rate for fixed, exhaustive search, and adaptively-selected reference frame schemes.

reference frame should not be heuristically pre-selected. This conclusion is further substantiated by Table 4.4, which provides the number of times a relative reference frame location yielded the best, second best, or worst prediction PSNR.

The adaptive algorithm selected the best reference frame for the majority of the frames in this sequence. In fact, the loss in prediction PSNR from using the adaptively selected frame, compared to the optimal reference frame, was greater than 0.5 dB for only 27 out of the total 270 frames. For clarity, we have re-plotted the prediction PSNR over the last 100 frames in Fig. 4.10b.

Results similar to these were obtained for the *Toy Train*, *Piano* and *Manege* two-view sequences. In each sequence, the frames in the dependent view were predicted using one of the temporally-closest reference frames. The average per-frame prediction PSNR and coded sequence bit rate for the dependent view are shown in Table 4.5, where the reference frames were either fixed, selected through exhaustive search, or selected using our adaptive algorithm. The results given for the fixed scheme are the combination of the results for all three possible reference frames. We note that for the *Manege* sequence the adaptive technique actually had a slightly lower bit rate than the exhaustive search method. This is most likely due to the fact that the prediction distortion was used as the selection criteria for the exhaustive method. A reference frame may then have yielded a lower prediction distortion yet another reference might have required fewer bits to encode.

The final results that we will examine illustrate the complexity reduction of our occlusion-based technique over an exhaustive search. The three-view *Cement Mixer* sequence was coded using View 0 as the independent view. The single-step predictions for the dependent Views 1 and 2 were obtained using reference frames in Views 0 and 1, respectively. The frames in the dependent views were predicted using two reference frames, where a maximum delay of one temporal period between the desired and reference frames was allowed. The set of possible reference frames for

	Exhaustive Search for Optimal Reference Frames					
	$F(0,n\text{-}1)$	$F(1,n\text{-}1)$	$F(2,n\text{-}1)$	$F(0,n)$	$F(1,n)$	$F(0,n+1)$
$F(1,n)$	0	210	210	0	N/A	0
$F(2,n)$	0	81	210	0	129	0

a.

	Adaptive Selection of Reference Frames					
	$F(0,n\text{-}1)$	$F(1,n\text{-}1)$	$F(2,n\text{-}1)$	$F(0,n)$	$F(1,n)$	$F(0,n+1)$
$F(1,n)$	4	210	198	6	N/A	2
$F(2,n)$	0	0	210	0	210	0

b.

Table 4.6: Number of times a relative reference frame location was selected for the prediction of the two dependent views, $F(1,n)$ and $F(2,n)$, a) exhaustive search technique, b) occlusion-based, adaptive technique.

this situation are given by Eq. (4.2). For the adaptive algorithm, the occlusion measure of the reference frame used for the generation of the single-step prediction was weighted by 0.95 to avoid an additional frame prediction when the amount of occlusion in the single-step reference was only slightly more than the minimum-occlusion reference.

The compression simulation was performed on a Silicon Graphics workstation, with a single RS4400, 174 MHz processor. The average processing times to select the two best reference frames and perform the final prediction of the desired frame for the exhaustive search technique was 128.4 seconds, while our technique required only 64.4 seconds per frame. This reduction in processing complexity did not result in a substantial loss in rate-distortion performance – the adaptive method achieved prediction PSNR and coded bit rates within 0.1% of those for the exhaustive search technique for this sequence.

Table 4.6 provides the number of times a relative reference frame location was selected for the two dependent views for both the exhaustive and adaptive techniques. Each row sums to 420 frames since the sequence consists of 210 frames per view and two reference frames were used for the prediction of each frame. These tables yield identical conclusions: the intra-view references are almost always one of the two best references, Views 1 and 2 are highly similar, and neither dependent view is very similar to View 0. These results again illustrate our point, since based purely on the camera configuration, one would likely assume that $F(0, n)$ would be one of the best two reference frames for $F(1, n)$.

4.7 Summary

In this chapter, we have presented a new algorithm for the adaptive selection of the best possible reference frame for the predictive coding of multi-view signals. The selection is based on an estimate of the amount of occlusion between the desired frame and each possible nearby reference frame. We utilized the relationship between occlusion and displacement vector discontinuity, and estimated the amount of occlusion from the variance of an augmented, composite displacement vector field. A set of simple tools to detect unoccluded regions were developed, which yielded highly reliable, bidirectional displacement estimates for the variance calculation. The composite fields were obtained through vector field addition of single-step fields, and did not require a costly displacement estimation process for each candidate reference frame.

The experimental results validated our hypothesis that the reference frame should not be preselected. By using the optimal reference frame, the overall sequence bit rates were reduced from 10-30% compared to fixed reference frame schemes, while maintaining the same level of reconstructed image quality. Our algorithm achieved rate-distortion performance within 1% of an exhaustive search method and consumed substantially fewer processing cycles. The major cost of this technique is that it requires approximately 2-3 times the storage of either a fixed or exhaustive search method.

Future work includes refining the estimate of a particular reference frame's performance to further improve the accuracy of the selection. We speculate that the adaptive algorithm occasionally selected the incorrect reference frame due to the fact that all occlusions are not equal – a region may be occluded between two views yet a satisfactory prediction for this region is possible. This would most likely occur when the occlusion has relatively uniform intensity. Basing the selection solely on the amount of occlusion does not adequately handle such a situation. We propose that incorporating a measure of the uniformity of invalidated image regions into the selection criteria may provide a solution to this problem.

Chapter 5

Restoration-based Residual Coding

In this chapter, we explore the low bit rate coding of the residual between the original and frame-based predicted images. We begin by illustrating, both in theory and in practice, how the complexity and bit rate constraints of the prediction process cause the predicted image to be sub-optimal, in the sense that it is correlated with the residual. We then discuss various source coding techniques and demonstrate their shortcomings with respect to this correlation. This leads to our novel class of encoders, which view the decoding process in terms of *restoring* the residual image from the prediction.

The basic functionality of the restoration-based residual coder resembles that of a predictive vector quantizer. However, instead of utilizing a fixed, or slowly-varying prediction filter, a unique restoration transform (vector predictor) is coupled with each reproduction vector. Each transform/vector pair, or restoration operator, compensates for a class of degradations due to the non-stationarity of image intensities along displacement trajectories and the practical limitations of the prediction process. Encoding is performed by mapping the residual vector to the index of the restoration operator that minimizes the distortion between the original and reconstructed vector.

Theoretical comparisons are made with comparable vector quantization implementations, which are, in fact, special, and generally sub-optimal, cases of restoration-based coding. Simulations on our multi-view signals yield gains of over 1 dB in the low bit rate region of 0.1 to 0.2 bits per pixel. While not quite as impressive, gains also are achieved over a common DCT-based coder. Due to its relatively high encoder complexity, this technique is most applicable for systems with severe bit rate constraints, such as multi-view signals.

5.1 Residual Coding

From a complete prediction of the current frame to be encoded,[1] the residual encoder attempts to either: 1) reconstruct the original image to a desired level of quality using the minimum number of bits (i.e., fixed-quality compression), or 2) reconstruct the original image to the best level of quality using at most a specified number of bits (i.e., fixed-rate compression). This operation constitutes the lossy compression portion of the hybrid encoder. The rate-distortion performance of a residual encoder is related to the quality of the prediction and the technique's ability to remove redundancy present in the given signal.

We are interested in residual encoders operating in the low bit rate region of less than 0.25 bits-per-pixel. This restriction is due to our desire to ensure that the coded multi-view signal has a bit rate demand commensurate with the benefits afforded by multiple views of the scene; i.e., severe compression of these signals is required to allow for their practical use. To achieve acceptable reconstructed image quality at these low rates, all correlations present in the signal must be exploited. Previously reported residual coding implementations for image sequences typically exploit the spatial correlation between neighboring pixels in the displacement-compensated frame difference. We hypothesize that other forms of correlation are present in the information available at both the encoder and decoder, which should be utilized to improve the residual coding performance.

In the following section, we describe how the non-stationarity of image intensities and the practical constraints on the prediction process result in significant correlation between the predicted and residual images. We then examine conventional residual encoder implementations, and speculate as to why these techniques traditionally neglect this correlation. Finally, we present our solution to the problem of low bit rate residual coding within a hybrid encoder framework, which is based on a combination of vector quantization and image restoration theory.

5.2 Predicted - Residual Image Correlation

To achieve superior rate-distortion performance, we assume that the algorithm presented in the previous chapter was used to select the minimum-occlusion reference frame for the prediction of

---

1. Conventionally, the input to the residual encoder is the difference, or error, between the original and predicted images (see Fig. 1.2). To provide insight into the functionality of this operation, we will consider the prediction as the input and combine the subtraction step into the residual encoding process.

the frame to be encoded in the multi-view signal. This assumption allows us to neglect the effect of occlusions on the prediction process, and to focus on the residual coding of unoccluded regions.

Recall our image model for unoccluded regions described in Section 3.1, where we represented pixel intensity along a displacement trajectory using a Gauss-Markov process. A spatiotemporally-varying correlation coefficient was incorporated into the conventional model to handle lighting and camera pick-up variations, which are typical in multi-view signals. Therefore, we explicitly assumed a non-stationary source model for image intensities. For convenience, we have re-written the image model given in Eq. (3.1), below:

$$I(u, v, m, n) = a_{00}(u, v) \cdot I(u - d_u(u, v), v - d_v(u, v), m_{\text{ref}}, n_{\text{ref}}) + w_{\text{un}}(u, v) \tag{5.1}$$

The frame-based hybrid encoder attempts to exploit the inherent memory of this source by predicting a pixel intensity from a previously decoded image frame via the estimation of the correlation coefficient and the displacement vector.

Practical limitations imposed on the prediction process affect the quality of the predicted image and, hence, the performance of the residual encoder. These limitations include constraints on the number of bits allocated to describe the prediction, the processing time allowed to perform the prediction, and the introduction of noise due to the lossy coding of the residual and the feedback structure of the hybrid coder. We next describe these limitations and their effect on the quality of the predicted image or, equivalently, the "randomness" of the resulting residual image.

As discussed in Section 3.1.3, a trade-off exists between allocating more bits to describe the prediction process or more bits to perform the residual coding stage. Ideally, a joint rate-distortion minimization would be performed to calculate the optimal bit allocation between these two processes. A technique to perform this optimization was presented in [82], for the case of lossless video coders. Here, an approximation of the total rate for encoding the current frame was related to the resolution of the estimated motion vectors. This optimization is decidedly more complex for lossy encoders. Therefore, approximations of the image model often are used, which result in a relatively constant number of bits to describe the prediction process.

This is the case for the common fixed-sized, block-based displacement estimation techniques, where a single displacement vector, estimated to a specified level of accuracy, is transmitted for each identically-sized, nonoverlapping block in the current frame [1, 2, 3, 4]. These prediction techniques have well-defined rate and complexity costs, which are related to the displacement range search area. Slight variations in the prediction bit rate are possible due to the entropy coding of the displacement vectors.

It is also customary to use a spatially-invariant prediction coefficient that approximates the correlation coefficient in Eq. (5.1) over an entire image or a class of images [1, 2, 3, 4, 71]. For either case, the overhead in terms of the complexity to estimate the prediction coefficient and the number of bits to transmit this quantity are inconsequential with respect to the displacement estimation and residual encoding operations.

While the assumptions of a spatially-invariant correlation coefficient and constant displacement over a fixed-sized region of pixels are often justified, they are nonetheless approximations – depending on the deviation of the actual parameters from these approximations, the prediction operation might yield a very poor estimate of the original image. By using more accurate approximations of the image model, it may be possible to generate a prediction that jointly requires fewer bits to describe and is a more faithful reproduction of the original image. However, the overall cost of this more elaborate operation must be considered; i.e., we must ensure that the prediction can be performed within the timing constraints of the encoder.

For example, consider a frame-based predictor that assumes constant displacement over an arbitrarily-shaped region of contiguous image pixels, and performs an exhaustive search on the size and shape for the optimal image regions. The minimum rate-distortion prediction may be obtained by this operation since it possess greater leeway in the manner in which the prediction is formed, and it assumes a potentially more accurate model of the true structure of the scene. However, an excessive amount of time might be required to examine all of the possible permutations afforded by arbitrarily-shaped image regions. To reduce the processing delay of this operation, some structure must be imposed on the possible image regions that are predicted using a single displacement estimate; this, in turn, reduces the quality of the prediction. Therefore, a trade-off also exists between minimizing the processing delay and the rate-distortion cost of the prediction process.

The final limitation of the prediction process that we will consider deals with the feedback nature of the hybrid coder. Due to the lossy compression of the residual encoding process, the reconstructed image will not be identical to the original image. This quantization noise will propagate through the sequence since the prediction must be performed in a closed-loop structure, i.e., the prediction is generated from a previously decoded reference frame. The effect of this discrepancy is particularly troublesome in very low bit rate regions, where coarse quantization of the residual image results in a violation of the additive noise model often assumed by recursive, predictive coding schemes [29].

Each of these practical limitations of the prediction process result in the same conclusion: optimal prediction of unoccluded pixel intensities, in the sense that the residual exactly equals the innovations sequence, is impossible. The residual image, therefore, is not an uncorrelated white-noise process, and instead is correlated with the signal. To formalize this correlation, we next present the equations that describe the predicted and residual images, where we include the non-ideal prediction quantities due to coarse quantization, and the bit rate and complexity cost constraints imposed on the prediction process.

The prediction of a pixel in the original image, from a reconstructed reference frame, is obtained from a slight modification of the standard relationship of Eq. (3.3):

$$\hat{I}(u, v, m, n) = \bar{I}(u - \hat{d}_u(u, v), v - \hat{d}_v(u, v), m_{\mathrm{ref}}, n_{\mathrm{ref}}) \tag{5.2}$$

The quantity $\bar{I}$ represents the reconstructed pixel intensity at the corresponding point in the reference frame, and is related to the actual pixel intensity by,

$$\bar{I}(u, v, m_{\mathrm{ref}}, n_{\mathrm{ref}}) = I(u, v, m_{\mathrm{ref}}, n_{\mathrm{ref}}) + q(u, v) \tag{5.3}$$

where q is the quantization noise of the residual encoding process. The residual pixel intensity is simply the difference between the original and predicted values:

$$e(u, v, m, n) = I(u, v, m, n) - \hat{I}(u, v, m, n) \tag{5.4}$$

For clarity, we respectively define the actual and estimated pixel locations of the corresponding point in the reference frame as,

$$\boldsymbol{p} = \begin{bmatrix} u - d_u(u, v) \\ v - d_v(u, v) \end{bmatrix} \text{ and } \hat{\boldsymbol{p}} = \begin{bmatrix} u - \hat{d}_u(u, v) \\ v - \hat{d}_v(u, v) \end{bmatrix} \tag{5.5}$$

Substituting the appropriate quantities into Eq. (5.4) yields the somewhat unwieldy expression of the residual intensity in terms of the uncoded reference frame:

$$e(u, v, m, n) = a_{00}(u, v) \cdot I(\boldsymbol{p}, m_{\text{ref}}, n_{\text{ref}}) + w_{\text{un}}(u, v) - I(\hat{\boldsymbol{p}}, m_{\text{ref}}, n_{\text{ref}}) - q(\hat{\boldsymbol{p}}) \tag{5.6}$$

Obviously, if $a_{00}(u, v) = 1$, $\boldsymbol{d}(u, v) = \hat{\boldsymbol{d}}(u, v)$ and $q(u, v) = 0$ over the entire image, the residual is merely the innovations sequence of the Gauss-Markov image model. In practice, though, this is not possible. It is clear then that the predicted and residual images are correlated; in other words,

$$E\{e(u, v, m, n) \cdot \hat{I}(u, v, m, n)\} \neq E\{e(u, v, m, n)\} \cdot E\{\hat{I}(u, v, m, n)\} \tag{5.7}$$

where $E\{\ \}$ denotes expectation. The degree of this correlation is inversely related to the quality of the prediction.

The practical significance of this correlation is illustrated in Fig. 5.1. The original frame *F*(0,45) and its decoded reference frame *F'*(1,45), from the *Manege* sequence, are shown in Figs. 5.1a and 5.1b, respectively.[2] The reference frame was coded using a rate-controlled, DCT-based residual coder, with a constant coded bit rate of 4 Mbps per view. The original image was predicted from the reference frame using a unity prediction coefficient and a fixed-sized, block-based displacement estimation technique (exhaustive search) with 16×16 pixel blocks and half-pixel accuracy for the displacement vector components. Figures 5.1c and 5.1d respectively contain the frame-based prediction and the absolute error between the original and predicted image for this frame. While locally the error image appears quite random, viewed globally the basic structure of the scene becomes apparent.[3]

Intuitively, if we, as observers, can recognize the structure of the scene from the residual, redundancy exists between the predicted and residual images. Since the prediction is available at the decoder, we hypothesize that this information can be used to achieve superior rate-distortion performance for the residual encoding process. Our task then is to formulate a method that efficiently exploits this redundancy and requires acceptable increases in complexity and storage costs.

---

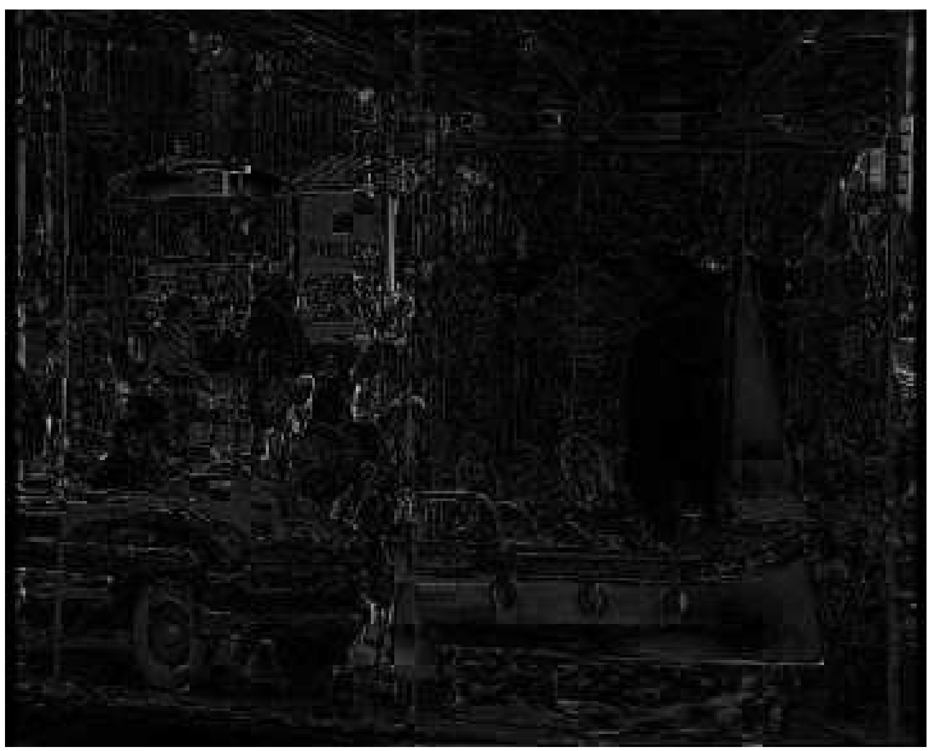
2. While the inter-view reference frame is not the minimum-occlusion reference of *F*(0,45), this reference provides a more powerful demonstration of the practical constraints on the prediction process since significant lighting variation exists between the two views.
3. The dynamic range of the residual image shown here suffers from the limitations of the printing process. When displayed on a high-resolution display, scene objects are easily recognized.

a.

b.

Figure 5.1: Illustration of predicted-residual image correlation for *Manege* sequence, a) original $F(0,45)$, b) decoded reference $F'(1,45)$, c) predicted image, d) absolute error image.

c.



d.

Figure 5.1 (cont.):

5.3 Conventional Techniques

In this section, we review common techniques for encoding the residual image within a hybrid encoder framework. In all implementations known to the author, only the spatial correlation within the residual image is exploited; i.e, the correlation between the predicted and residual images is neglected. We speculate that this sub-optimal approach is due to the fact that a stationary image model is often valid for single-view signals, and that these coders typically operate in more moderate-to-high bit rate regions. For these situations, the predicted-residual image correlation will be negligible, and the gain achieved by exploiting this redundancy will probably not justify the increased computational effort. However, these assumptions are invalid for our problem of low bit rate coding of multi-view signals; thus, it will be advantageous to exploit this correlation. In the following discussion, we highlight concepts that will be potentially useful for our task.

5.3.1 Transform-based with Scalar Quantization

A popular method for compressing the residual image is a transform-based technique [1, 2, 3, 4, 5, 22, 60, 78]. Here, the residual image data is transformed to a generalized frequency domain, with the goal of decorrelating the signal and compacting most of the energy in the signal into a small number of transform coefficients. A combination of both scalar quantization and entropy encoding then is applied to these coefficients. To reduce the computational complexity of the transformation, these techniques typically are applied to relatively small-sized image blocks, where the size of the block used is a trade-off between spatial and frequency resolution [79].

The Karhunen-Loève Transform (KLT) is known to achieve optimal decorrelation and energy compaction for a class of image sources [18]. However, the KLT has seen limited use in residual coding applications since the statistics of the image source often are not known a priori, and the computation and subsequent transmission of the KLT for a particular image is prohibitive.

The common approach then has been to use a non-optimal, data-independent transform that is related to the statistics of all image sources. For example, the Discrete Cosine Transform (DCT) approximates the KLT for highly correlated sources [42, 80]. This is the transform used in the residual coding stage of the MPEG and H.26x video compression standards. Although the residual image often contains considerable contrast and abrupt edges (i.e., it has relatively low spatial correlation), the DCT repeatedly has been shown to achieve adequate, mid-bit rate performance. Also, fast implementations of the DCT exist [57, 80], and it is possible to “piggy-back” off of existing hardware used to perform DCT-based intraframe compression [60].

A solution to some of the difficulties associated with using the KLT recently was presented in [24], where a finite set of transforms were designed that approximate the optimal transform for a broad class of image sources, i.e., it is *universal* in nature. While this technique was applied to intraframe coding, it should be easily extended to the residual coding problem. We make the observation that this algorithm might benefit from the correlation between the predicted and residual images. For example, one can imagine a situation were the particular transform used for a block of pixels in the residual image is based on the statistics of the corresponding block in the predicted image. While we will not explore this thought further, we will revisit the concept of universal coders in our novel class of restoration-based residual encoders (see Section 5.4).

5.3.2 Vector Quantization

Another common set of techniques for the compression of images and video are known as vector codes or vector quantizers (VQ) [19, 35, 36, 39, 50, 58, 69, 85]. From the classic result of Shannon, by coding blocks or vectors of source samples as a unit rather than individually, the theoretical limit on compression performance can be approached arbitrarily closely, in the sense of minimizing the average distortion for a specified bit rate [88]. The bound is approached as the dimensionality of the vector goes to infinity. This result holds even if the source is memoryless.

Applying this technique to the problem of residual encoding, the residual image first is partitioned into a set of nonoverlapping image blocks, which are then listed in vector form. Encoding is performed by mapping each vector within this set to a finite index. The indices then are transmitted to the decoder, and the quantized residual vector is decoded by mapping the received index to a reproduction vector drawn from a finite reproduction alphabet, or codebook. The optimum encoding rule is the nearest-neighbor rule, in which a particular index is selected if the associated code vector yields the minimum distortion over all reproduction vectors in the codebook.

For N-dimensional vectors and a codebook of M reproduction vectors, the compression ratio of the vector quantizer is $(\log_2 M)/N$, assuming that the indices are not entropy coded. The vector quantization codebook frequently is designed from a set of representative training vectors using a clustering algorithm, such as the generalized Lloyd algorithm (GLA) [54].

In its most basic form (full-search vector quantization), the codebook is unstructured, and the encoding process entails a full-search for the minimum-distortion reproduction vector. For high bit rates, this search is prohibitive. For example, if a vector contains 16 pixels and a rate of 1 bit-per-pixel is desired, the search must be performed over 65536 reproduction vectors. Conversely, the decoder has a very simple implementation – the quantized vector is obtained by a simple table look-up based on the received index.

Numerous extensions of the basic vector quantizer have appeared in the literature. These techniques attempt to either improve the performance of the coder, reduce its complexity, or both. We briefly describe a sampling of these variants, which will prove useful in the latter sections of this chapter.

A tree-structured vector quantizer (TSVQ) constrains the codebook to consist of a tree of reproduction vectors [50, 58]. A balanced binary tree typically is employed (i.e., a tree with all of its terminal nodes at the same layer), and only a sequence of $(\log_2 M)$ binary searches is performed, greatly reducing the encoding process compared to the full-search VQ. The process of stepping through the tree-structured codebook can be viewed as a "successive approximation" of the input vector. Due to the constrained codebook structure, a fixed rate TSVQ generally will have average distortion higher than that of a comparable fixed rate full-search VQ.

Prior to entropy coding, the indices of a balanced TSVQ have fixed rate. A variable rate code can be obtained if the balanced tree is replaced by a pruned tree, where the terminal nodes lie at varying depths [16]. The number of bits to encode an index then will depend on the particular reproduction vector selected. Optimal pruning of the balanced tree is performed using the generalized BFOS algorithm [12]. The two variations of this algorithm trade either average rate (length-pruned tree-structure, PTSVQ) or average entropy (entropy-pruned tree-structure, EPTSVQ) for average distortion. Pruned tree-structured VQ implementations typically achieve superior rate-distortion performance over full-search VQ, while maintaining the reduced complexity feature of TSVQ.[4]

In the previously described VQ implementations, each vector is coded independently of previously encoded vectors, and are, therefore, referred to as memoryless VQ. Finite-state vector quantization (FSVQ) and predictive vector quantization (PVQ) are capable of exploiting inter-vector correlation [32, 39, 69]. As the name implies, FSVQ consists a finite number of states, where a distinct and relatively small codebook is associated with each state. The current state (uniquely available at both the encoder and decoder) is a function of the previous state and the previously encoded vectors, and the input vector is quantized using the corresponding state codebook. A PVQ scheme

4. Another approach to achieving a variable rate vector quantizer, termed entropy-constrained vector quantization (ECVQ), jointly optimizes an unstructured vector quantizer and a noiseless variable rate coder [17]. While this technique has the potential for optimal rate-distortion performance, it possess a very high codebook design complexity.

predicts the vector from previously decoded vectors and vector quantizes the residual, where the vector predictor is designed for optimal global performance (i.e, is fixed) or is allowed to vary slowly. PVQ can be interpreted as an "infinite-state" vector quantizer, where both the state and state codebook are a function of the previous vector. The design of a finite-state or a predictive vector quantizer respectively involves the joint optimization of either the next-state/state codebook or the predictor/residual-codebook.

An interesting, nonlinear predictive VQ (NLPVQ) technique was presented in [36] for the compression of multispectral imagery. A single spectral band is extracted from the multi-spectral imagery, and is defined as the *feature* band. Vectors within the feature band are coded using a standard vector quantizer. The quantized feature vector takes on a finite set of possible values, namely, the reproduction vectors of the VQ codebook. An optimal (mean-square error) nonlinear prediction of a corresponding vector in another spectral band then is generated from the conditional mean of this vector given that the feature vector was mapped to a specific codeword. Finally, if the prediction error for the vector in the second band is greater than a threshold, the residual vector is encoded by another VQ encoder. We note that this process is a "zero-rate" prediction; i.e., the prediction is uniquely defined by the quantized feature vector and the previously stored conditional means.

While not explicitly applied to the residual image coding task, one can envision how the predicted-residual image correlation might be exploited by NLPVQ. Consider a vector within the residual image and a corresponding vector in the predicted image that possesses some correlation with the residual vector. The predicted vector is analogous to the feature vector in NLPVQ, in that information of the residual vector, available from the predicted vector, is utilized to generate a superior representation of the residual vector.

We hypothesize that the NLPVQ paradigm can be improved by explicitly using knowledge of the original image to adaptively reconstruct the residual from the given prediction. This forms the basis of our novel concept for residual coding that aims at making the best possible use of the predicted image to reproduce the residual or, equivalently, original image.

5.4 New Approach: Restoration-based Coding

We begin our development of restoration-based coding by viewing the decoding process in terms of an operation performed on the predicted image, which is a degraded version of the original image. A measure of the encoder's performance must be a combination of how succinctly this operation can be described to the decoder and how well this operation reconstructs the original

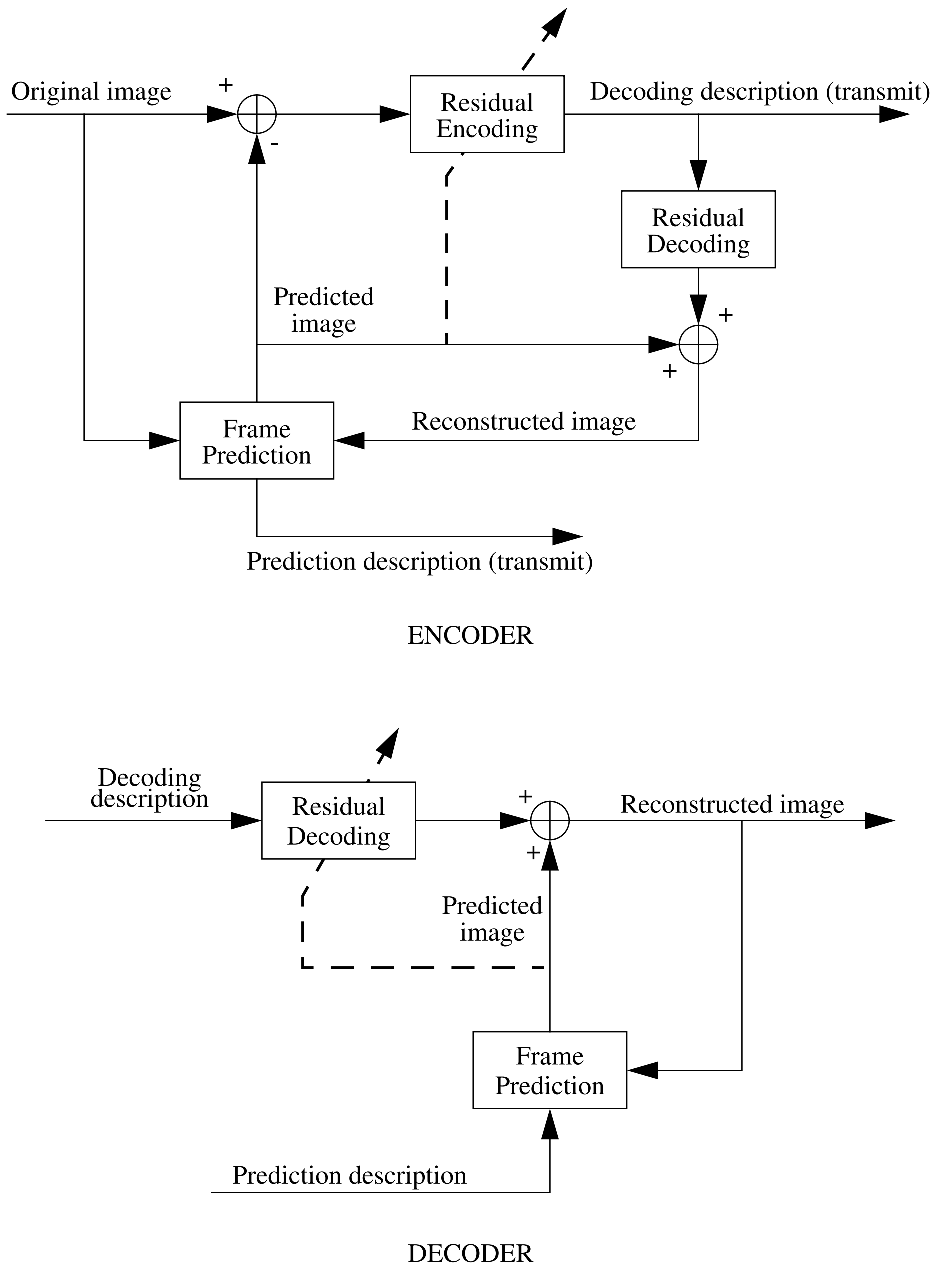


Figure 5.2: Frame-based hybrid encoder and decoder structures with “prediction-directed” residual coding.

image. As discussed in the previous section, this operation traditionally has amounted to the addition of a quantized (distorted) representation of the residual image to the prediction; the decoding description merely contains the information needed to generate the quantized residual image, which is obtained independently of the predicted image.

Our approach to exploiting the predicted-residual image correlation is essentially a "prediction-directed" residual coder. Figure 5.2 illustrates this concept on a modified hybrid coder structure, where the dashed-line represents the dependence of both the residual encoder and decoder on the predicted image.

We wish to find the optimal operation, in some sense, to be performed on the prediction that *restores* it to the original image. Since the residual image is simply the difference between the original and predicted images, we can, equivalently, formulate this task in terms of finding the optimal operation that restores the residual image from the prediction. The latter formulation is more consistent with the conventional hybrid coder structure; therefore, we will develop our technique within this context. To reduce the computational complexity, the operation developed to restore the residual image will be applied to the prediction in a piecewise manner.

We speculate that the a wide variety of degradations present in the predicted image, with respect to the residual image, can be represented by a finite set of *degradation classes*. If these classes can be quantified, the coding problem can be framed in terms of specifying the operation to "invert" the particular degradation that occurred at each image region. However, it would be very difficult to define these classes a priori; we do not know what degradations are important, in an information-theoretic sense. Therefore, in analogy to vector quantization, we design a codebook of *restoration operators,* based on a representative set of predicted-residual image pairs. This process is performed "off-line" and the codebooks are stored at both the encoder and decoder. Each operator is designed to optimally reconstruct the residual for a specific degradation class.

For each segment of the residual image, the encoder searches through the codebook for the restoration operator that, when applied to a corresponding region in the predicted image, minimizes the distortion between the original and restored residual image region. The decoding description is merely the index of the selected restoration operator, which then is applied to the copy of the predicted image available at the decoder. In the standard fashion, the decoded residual is then added to the prediction to complete the reconstruction of the original image frame. We call this approach *restoration-based residual coding*.

From this description, this process obviously is closely related to the problem of restoring an image from an observed, degraded versions of the original [21, 84]. In image restoration, only the degraded image and a model of the degradation mechanism are available.[5] A technique similar to our approach of applying a set of restoration operators to the degraded image in a piecewise manner, termed Vector Classified Adaptive Filtering (VCAF), was presented in [84]. Here, the selection of the restoration operator was based on a priori knowledge of typical image statistics in the form of a classification codebook.

The basic concept of VCAF is essentially equivalent to that of NLPVQ, discussed in Section 5.3.2, where the quantized feature vector is replaced by the classification codebook. Restoration-based coding differs from these approaches in that it explicitly uses the original image, available at the encoder, to "direct" the restoration (prediction) operation; VCAF and NLVPQ rely solely on the image statistics for adaptation and do not utilize the actual desired image to improve their performance.

Our approach to hybrid coding of image sequences can be interpreted as a two-stage predictor – the first stage generates the predicted image, and the second stage predicts (restores) the residual from the initial frame-based prediction. The justification for this method is based on the existence of predicted-residual image correlation. Ideally, these two predictions would be combined, and only the prediction description would need to be transmitted to the decoder (see Fig. 5.2). However, this would require an enormous search space; the minimum-distortion prediction would need to be searched for over all possible displacement-vector/restoration-operator permutations. While the decoupling of these steps results in generally sub-optimal performance, restoration-based coding is still able to efficiently exploit the inherent redundancy between the predicted and residual images.

We next describe the full encoder and decoder structure in detail, including the generation of the restoration operators. To simplify the discussion, we first present the scalar version (i.e., restoration of a single pixel in the residual image), then extend this to blocks or vectors. Theoretical interpretations and comparisons with vector quantization are made, and we extend some of the useful VQ variants to the vector restoration-based coder (VR).

5.4.1 Scalar Restoration (SR)

To simplify the notation, we respectively denote the predicted, original, and residual images by the $(U \times V)$ matrices $\boldsymbol{X}$, $\boldsymbol{Y}$ and $\boldsymbol{Z}$ where $\boldsymbol{Z} = \boldsymbol{Y} - \boldsymbol{X}$. Consider a pixel at (u, v) in the residual image,

---

5. For the task of blind deconvolution, only limited knowledge of the degradation mechanism is available. See [53] for a tutorial on this subject.

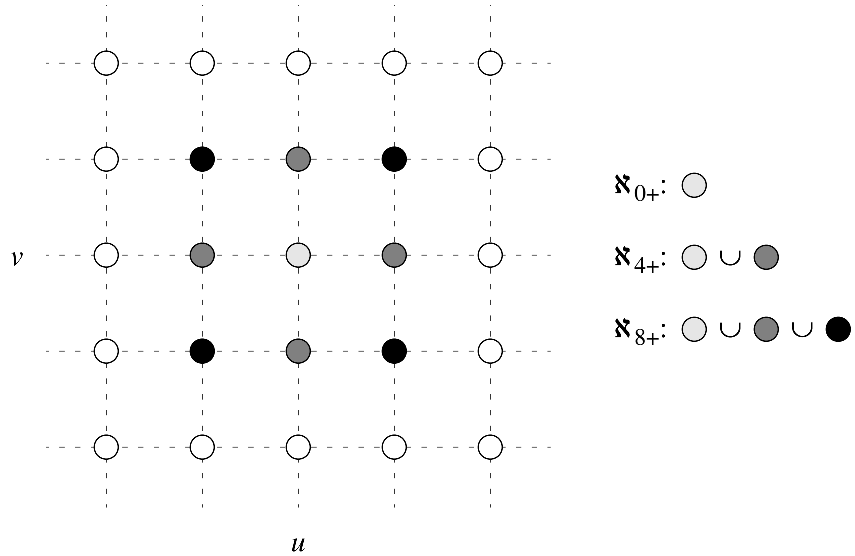


Figure 5.3: Typical pixel neighborhoods in predicted image for the restoration of pixel location (u, v) in the residual image. Each neighborhood is denoted by the union of shaded circles.

given by z, that is to be encoded using information in the displacement-compensated predicted image. In this discussion, scalars are denoted by lower-case letters, vectors by bold, lower-case letters, and matrices by bold, upper-case letters.

In general, the entire predicted image may be used to encode this pixel; in practice, a subregion in $\boldsymbol{X}$ is extracted consisting of all samples that contain information about z. The region then is ordered into a k-dimensional vector, denoted by $\boldsymbol{x}$. We note that there is no loss of optimality because $\boldsymbol{x}$ contains all of the information there is in $\boldsymbol{X}$ regarding the value z. Typically, $\boldsymbol{x}$ will consists of the set of k-closest pixels, in the sense of euclidean distance, to the corresponding image location of z in the predicted image, where the value of k depends on the accuracy of the displacement estimate (see Eq. (5.6)) This set of pixels is similar to the pixel neighborhood described in Section 3.1.2, with the difference that the center pixel in Fig. 3.1 may be used here since the complete predicted image is available. We denote this neighborhood by $\aleph_{k+}$ to indicate this difference. The neighborhoods for $k = (0, 4, 8)$ are shown by the union of the appropriately-shaded pixel locations in Fig. 5.3.

We assume that a codebook of M restoration operators, $f_i(\boldsymbol{x})$, are available at the encoder and decoder, and we denote this codebook by: $A(M) = \{f_i(\boldsymbol{x});\ i = 1, \ldots, M\}$. Encoding is performed by selecting the operator that yields the minimum-distortion prediction of z. In general, any distortion function may be used for the codebook search; however, for mathematical tractability, we will constrain our discussion to the mean-squared distortion measure. Therefore, the nearest-neighbor encoding rule is applied to generate the index i, such that,[6]

$$i = \arg \min_{1 \le j \le M} \left\{ \left\| z - f_j(\boldsymbol{x}) \right\|^2 \right\} \tag{5.8}$$

This process is essentially a mapping from the residual-predicted pair, $(z, \boldsymbol{x})$, to the finite index i. At the decoder, the received index specifies the desired restoration operation which is then applied to the predicted vector to yield, $\hat{z} = f_i(\boldsymbol{x})$. We next examine the design of the restoration codebook.

An iterative, clustering algorithm can be applied to design the codebook of restoration operations from a representative set of $(z, \boldsymbol{x})$ pairs, similar to the well-known Generalized Lloyd Algorithm (GLA) for vector quantization codebook design [54]. In each iteration, the training pairs are first partitioned by encoding with the current codebook. The restoration operation associated with partition i, (i.e., S_i), then is updated by computing the function that minimizes the partial mean-squared error over the partition. Specifically, we seek the predictor $\hat{z}$ that minimizes,

$$D_i = E\left\{ \left\| z - \hat{z} \right\|^2 \middle| \boldsymbol{x}; (z, \boldsymbol{x}) \in S_i \right\} \tag{5.9}$$

This is a standard least squares estimation problem, with a modification incorporating the known partition, and the solution is [42],

$$f_i(\boldsymbol{x}) = E\{ z | \boldsymbol{x}; (z, \boldsymbol{x}) \in S_i \} \tag{5.10}$$

---

6. The overall goal of the encoding process obviously is the minimization of the distortion between the original and reconstructed images. We may reformulate Eq. (5.8) to reflect this goal by substituting $(y - x)$ for z, which yields $\left\| y - (x + f_j(\boldsymbol{x})) \right\|^2$ for the bracketed-term. The restoration operator then can be defined as $g_j(\boldsymbol{x}) = x + f_j(\boldsymbol{x})$, i.e., the restoration-based coder operates directly on the predicted image to restore the original image. We note that these formulations achieve identical rate-distortion performance.

The operator calculation is performed for all codebook partitions, and this process is repeated until the overall distortion converges or reaches a specified level. Since the distortion is monotonically decreasing with each iteration, a locally optimal codebook is guaranteed by this design algorithm.

We observe that the calculation of the general restoration operators, $f_i(\boldsymbol{x})$, from Eq. (5.10) is impractical. For 8-bit images, the domain of $f_i(\boldsymbol{x})$ is $2^{8k} \approx 10^{2.4k}$; even small values of k would require an excessive computational effort to calculate the required conditional means. To overcome this difficulty, in this thesis, we restrict the restoration operators to be affine transforms:

$$\hat{z}_i = f_i(\boldsymbol{x}) = \boldsymbol{h}_i\boldsymbol{x} + c_i \tag{5.11}$$

where $\boldsymbol{h}_i$ is a k-dimensional row vector, c_i is a scalar, and the operator is parametrized by $(\boldsymbol{h}_i, c_i)$.

The optimal affine restoration transform for the i^{th} partition then is given by the Wiener-Hopf equation for non-zero mean sources [98]:

$$\boldsymbol{h}_i = \boldsymbol{r}_{z\boldsymbol{x}|i}\ \boldsymbol{R}^{-1}_{\boldsymbol{xx}|i} \text{ and } c_i = E\{z|i\} - \boldsymbol{h}_i E\{\boldsymbol{x}|i\} \tag{5.12}$$

where, $\boldsymbol{r}_{z\boldsymbol{x}|i} = E\{z\boldsymbol{x}^t|i\} - E\{z|i\}E\{\boldsymbol{x}^t|i\}$, $\boldsymbol{R}_{\boldsymbol{xx}|i}$ denotes the conditional auto-covariance matrix of $\boldsymbol{x}$, and each of the quantities in Eq. (5.12) is with respect to (i.e., conditioned on) the i^{th} partition. These values can be numerically estimated by computing time-averages over the partitioned training data.

Higher rate codebooks also are obtained in a manner analogous to that of vector quantization codebook splitting [54]. At initialization, we set $M = 1$ and define the initial codebook, $A(1) = \{(\boldsymbol{h}_1, c_1)\}$, as the solution of Eq. (5.12) for the entire training set. Given the codebook $A(m)$, we "split" each codebook entry $(\boldsymbol{h}_i, c_i)$ by perturbing c_i into the two close values $c_i + \varepsilon_i$ and $c_i - \varepsilon_i$. The new codebook then has $2M$ affine transformations:

$$A(M) = \{(\boldsymbol{h}_i, c_i + \varepsilon_i), (\boldsymbol{h}_i, c_i - \varepsilon_i);\ i = 1, \ldots, M\} \tag{5.13}$$

The $\boldsymbol{h}_i$'s are updated after the first iteration of the GLA on the new codebook. We set the perturbation constant equal to one-half the standard deviation of the difference between the actual and restored residual pixels, i.e., $\varepsilon_i = \frac{\sigma_i}{2}$, where:[7]

$$\sigma_i^2 = E\{\|z - \hat{z}\|^2\} = r_{zz|i} - \boldsymbol{r}_{z\boldsymbol{x}|i}\,\boldsymbol{h}_i^t \tag{5.14}$$

5.4.2 Vector Restoration (VR)

To realize the gains afforded by dimensionality [88], we develop the vector restoration-based coder, which is merely a generalization of the scalar case.

Here, the residual image is sectioned into small, nonoverlapping subimages containing N pixels, and the pixel values of the subimages are listed into the column vector denoted by $\boldsymbol{z}$. For each component of the residual vector, z_j, we extract the subregion in the predicted image consisting of all samples that contain information about z_j, and this region is ordered into the k_j-dimensional "partial" predicted vector, $\boldsymbol{x}_j$. We note that the partial predicted vectors, for all j, are not restricted to be equivalent or even have the same dimension. The partial predicted vectors are combined into the "complete" predicted vector, $\boldsymbol{x}$, where common terms are collected and retained only once in the meta-vector. The maximum dimension of this vector, K, is less than or equal to $\sum_{j=1}^{N} k_j$, and, more often, is approximately equal to N.

The residual vector is encoded in the same fashion as the scalar case using the nearest-neighbor encoding rule. The reconstructed residual vector using the i^{th} codebook entry is now given by,

$$\hat{\boldsymbol{z}}_i = \boldsymbol{H}_i\boldsymbol{x} + \boldsymbol{c}_i \tag{5.15}$$

where $\boldsymbol{H}_i$ is an $(N \times K)$ matrix, $\boldsymbol{c}_i$ is an N-dimensional column vector, and the codebook is denoted by: $A(M) = \{(\boldsymbol{H}_i, \boldsymbol{c}_i);\ i = 1, \ldots, M\}$.

Codebook design again is performed using the Generalized Lloyd Algorithm and a representative set of $(\boldsymbol{z}, \boldsymbol{x})$ pairs. For each partitioned training set, we calculate the optimal affine recon-

---

7. Note that $(z - \hat{z})$ has zero mean.

struction transform for each component of the respective residual vector by modifying Eq. (5.12) to indicate the dependence on the position j in the residual vector:

$$\boldsymbol{h}_{ij} = \boldsymbol{r}_{z_j \boldsymbol{x}_j | i} \, \boldsymbol{R}^{-1}_{\boldsymbol{x}_j \boldsymbol{x}_j | i} \text{ and } c_{ij} = E\{z_j | i\} - \boldsymbol{h}_{ij} E\{\boldsymbol{x}_j | i\} \tag{5.16}$$

where $\boldsymbol{h}_{ij}$ is a k_j-dimensional row vector. The vector $\boldsymbol{c}_i$ is obtained by simply stacking the N scalars, c_{ij}, into column-vector form. The specification of $\boldsymbol{H}_i$ from Eq. (5.16), requires a slightly more detailed description.

Each $\boldsymbol{h}_{ij}$ corresponds to a row of the matrix $\boldsymbol{H}_i$; however, $\boldsymbol{H}_i$ is not necessarily obtained by stacking the N row vectors $\boldsymbol{h}_{ij}$. Instead, the components of $\boldsymbol{h}_{ij}$ are placed at the locations in the j^{th} row of $\boldsymbol{H}_i$ according to the placement of the components of the partial predicted vector in $\boldsymbol{x}$; the rest of the row's entries are set to zero. Typically $\boldsymbol{H}_i$ is a sparse matrix. For example, if the $\aleph_{0+}$ neighborhood was used for each $\boldsymbol{x}_j$, only the main diagonal of $\boldsymbol{H}_i$ will be non-zero; for $\aleph_{4+}$, only the main diagonal and four off-diagonals will be non-zero; and so on. Also, since $\boldsymbol{H}_i$ is in general a shift-variant filter, it is not necessarily a block-Toeplitz matrix [42]. If $\boldsymbol{x}_j = \boldsymbol{x}$ (i.e., each partial predicted vector is identical), the auto-covariance matrix and its inverse need to be computed only once for the generation of $\boldsymbol{H}_i$ in Eq. (5.15).

Finally, codebook splitting is performed by appropriately modifying Eqs. (5.13) and (5.14) to reflect the extension of scalars to vectors and vectors to matrices; i.e., the perturbation vector ε_i splits the codebook entry parameterized by $(\boldsymbol{H}_i, \boldsymbol{c}_i)$, into $(\boldsymbol{H}_i, \boldsymbol{c}_i + \varepsilon_i)$ and $(\boldsymbol{H}_i, \boldsymbol{c}_i - \varepsilon_i)$.

Figure 5.4 illustrates the vector restoration encoder and decoder structures. We make two observations on this diagram: the vector formation blocks are different for the residual and predicted images, in that only nonoverlapping blocks are extracted from the residual image, and the predicted blocks may overlap; and the “restoration” and “minimize” blocks in the encoder constitutes the nearest-neighboring encoding procedure.

5.4.3 Interpretation and Comparison

Restoration-based coding is both *adaptive* and *universal* in nature [24, 85, 105]. Its adaptive feature is based on the ability to adaptively select the best restoration operator for a region in the residual image; basically, it is able to “track” non-stationarities of the source signal. We note that rapid fluctuations[8] in the source's statistics are handled by this approach. Provided that the set of images used to design the codebook is sufficiently diverse, multiple restoration operators also yield the universal characteristic, in the sense that this approach is able to a priori encode a large class of distinct sources. The restoration-based coder is essentially a locally linear (affine) – globally nonlinear, adaptive, shift-variant filter ("cluster-wise linear").

Scalar- and vector-restoration are, respectively, related to Adaptive Differential Pulse Code Modulation (ADPCM) and adaptive predictive vector quantization (APVQ) [50, 108]. In these techniques, the prediction and quantization steps are decoupled. The major contribution common to both SR and VR is that the adaptation does not require the transmission of any side information. This is achieved by coupling the adaptive predictor ($\boldsymbol{H}_i$) with the quantizer ($\boldsymbol{c}_i$); i.e., they are jointly specified by one index. This coupling also allows for the predictor and quantizer to be jointly optimized in the codebook design phase.

While techniques for adaptive coding without side information have appeared in the literature [36, 70, 85, 105], these methods base their adaptation on previously decoded signal; our approach utilizes the current signal. The decoder is able to follow this adaptation since it is explicitly described in the transmitted index.

Vector (scalar) restoration requires the same bit rate as non-adaptive PVQ (DPCM), but it overcomes the limitations of using a single prediction filter designed for optimal global performance. From Eq. (5.15), it is clear that PVQ is a special, and generally, sub-optimal case of VR, where $\boldsymbol{H}_i = \boldsymbol{H}$ (i.e., the predictor is fixed for all partitions). Therefore, at any given bit rate, VR outperforms or, in the worst case, is equivalent to PVQ in terms of average distortion. We quantify this improvement with the following analysis.

For simplicity, we assume that the source signal is partitioned into identical partitions for both VR and PVQ; that is, the vector z will be encoded with index i in both techniques. The average distortion for the signal is then given by $D = \frac{1}{M}\sum_{i=1}^{M} D_i$, where

$$D_i = E\left\{\|z - \hat{z}\|^2 \,\middle|\, \boldsymbol{x};\, (z, \boldsymbol{x}) \in S_i\right\} \tag{5.17}$$

8. In the sense that a large change of the source's statistics occur over a short duration window in time-space.

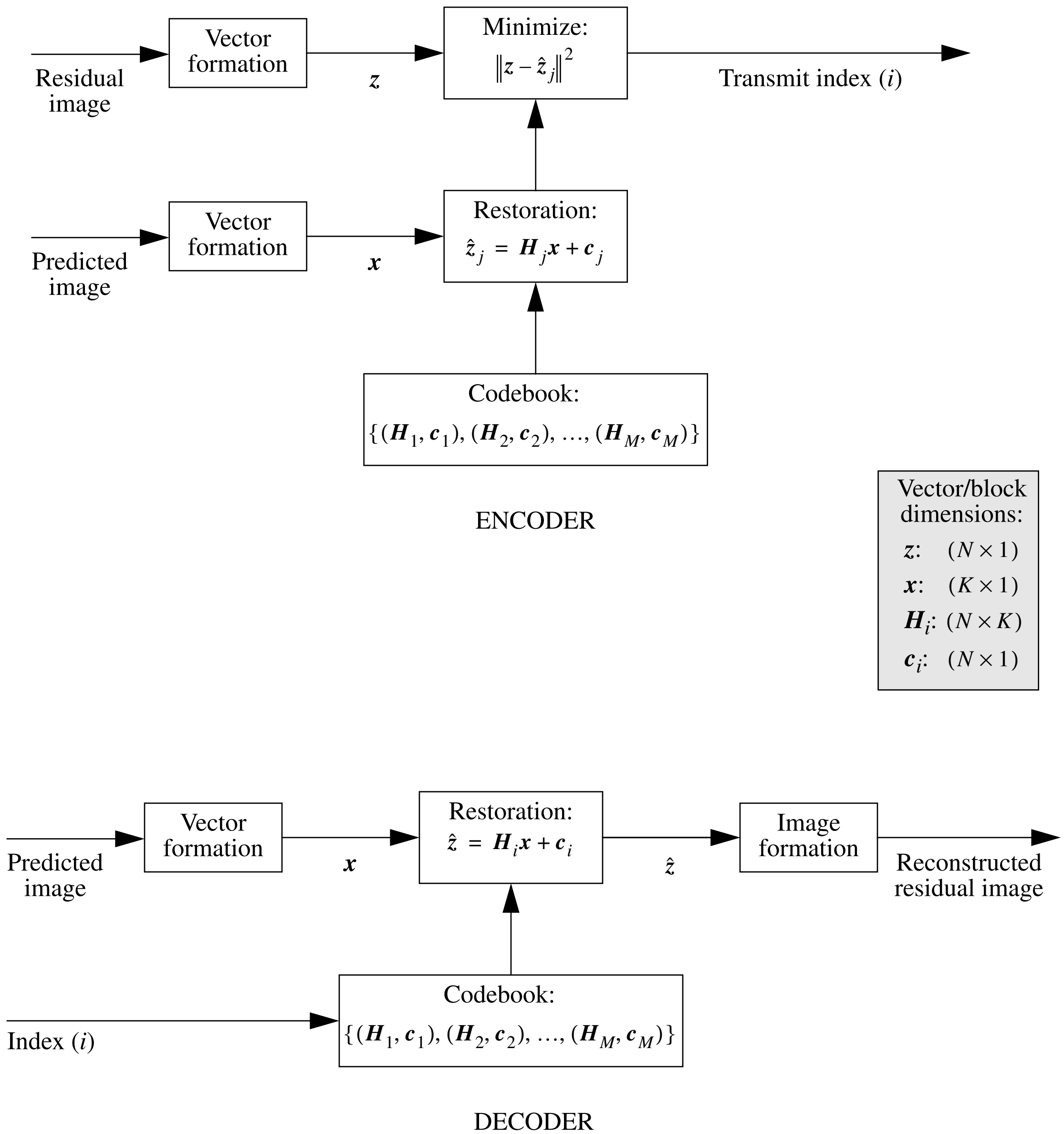


Figure 5.4: Vector restoration encoder and decoder structures.

is the distortion for partition i. Substituting $\hat{z}_j = \boldsymbol{h}_j \boldsymbol{x}_j + c_{ij}$ for PVQ[9] (i.e. fixed $\boldsymbol{H}$ for all partitions), and $\hat{z}_j = \boldsymbol{h}_{ij} \boldsymbol{x}_j + c_{ij}$ for VR, into Eq. (5.17) yields:

$$D_{PVQ|i} = \frac{1}{N} \sum_{j=1}^{N} \left\{ r_{z_j z_j|i} - 2\boldsymbol{h}_j \boldsymbol{r}_{z_j \boldsymbol{x}_j|i} + \boldsymbol{h}_j \boldsymbol{R}_{\boldsymbol{x}_j \boldsymbol{x}_j|i} \boldsymbol{h}_j^t \right\} \tag{5.18}$$

$$D_{VR|i} = \frac{1}{N} \sum_{j=1}^{N} \left\{ r_{z_j z_j|i} - \boldsymbol{r}_{z_j \boldsymbol{x}_j|i} (\boldsymbol{R}_{\boldsymbol{x}_j \boldsymbol{x}_j|i}^{-1}) \boldsymbol{r}_{\boldsymbol{x}_j z_j|i} \right\} \tag{5.19}$$

The average reduction in distortion, for this partition, realized by using VR as opposed to PVQ is then,

$$G_i = D_{PVQ|i} - D_{VR|i} = \frac{1}{n} \sum_{j=1}^{N} \left\{ r_{z_j \boldsymbol{x}_j|i} (R_{\boldsymbol{x}_j \boldsymbol{x}_j|i}^{-1}) r_{\boldsymbol{x}_j z_j|i} + \boldsymbol{h}_j R_{\boldsymbol{x}_j \boldsymbol{x}_j|i} \boldsymbol{h}_j^t - 2\boldsymbol{h}_j r_{z_j \boldsymbol{x}_j|i} \right\} \tag{5.20}$$

The minimum of Equation (5.20) occurs, and is equal to zero, when either $r_{z_j \boldsymbol{x}_j|i} = 0$, i.e., z_j and $\boldsymbol{x}_j$ are uncorrelated, or $r_{z_j \boldsymbol{x}_j|i} R_{\boldsymbol{x}_j \boldsymbol{x}_j|i}^{-1} = r_{z_j \boldsymbol{x}_j} R_{\boldsymbol{x}_j \boldsymbol{x}_j}^{-1}$, i.e., the local statistics equal the global statistics.

5.4.4 Extensions

Since the only functional difference between vector restoration and vector quantization is the substitution of restoration operators for reproduction vectors, we can easily apply all of the VQ codebook variations described in Section 5.3.2 to VR. As we will show in our cost analysis (Section 5.5), the complexity and storage costs of vector restoration can be substantially higher than either VQ or transform-based techniques; therefore, the reduced complexity techniques will be particularly useful.

9. Here, $\boldsymbol{h}_j = \boldsymbol{r}_{z_j \boldsymbol{x}_j} \boldsymbol{R}_{\boldsymbol{x}_j \boldsymbol{x}_j}^{-1}$ and $c_{ij} = E\{z_j\} - \boldsymbol{h}_j E\{\boldsymbol{x}_j\}$.

5.5 Cost Analysis

In this section, we provide the complexity and storage costs of the vector restoration residual coder. Comparisons are made with common DCT-based and vector quantization coding schemes. To facilitate a meaningful comparison, we assume that the block sizes for each of these techniques are identical – all common DCT-based techniques use 8×8, so $N = 64$ in the VR and VQ schemes. Due to the asymmetric nature of our approach, we quantify the relative costs for both the encoder and decoder structures.

5.5.1 Complexity

Since all techniques must subtract the predicted image from the original image to generate the residual, and perform the opposite operation at the decoder, we will not consider this step in our complexity analysis.[10]

An image block is encoded using VR by calculating the restored vector, $\hat{\boldsymbol{z}}_i$, and its distortion, D_i, for each candidate codebook entry. For the j^{th} pixel in the residual vector, this amounts to the calculation of both $\hat{z}_{ij} = \boldsymbol{h}_{ij}\boldsymbol{x}_j + c_{ij}$, which requires k_j multiplications and k_j additions, and $D_{ij} = (z_j - \hat{z}_{ij})^2$, which requires 1 subtraction and 1 multiplication. For simplicity, we assume that the partial predicted vectors all have the same dimension, i.e., $k_j = k$.

For full-search VR (unstructured codebook), the per-pixel encoder complexity of our technique is then,

$$C_{\text{VR-encoder}} = 2M(k+1) \tag{5.21}$$

where we recall M is the number of restoration operations in the codebook. For tree-structured vector restoration (TSVR), only a sequence of binary searches are performed, and the complexity is reduced considerably to:

$$C_{\text{TSVR-encoder}} = 4(\log_2 M)(k+1) \tag{5.22}$$

---

10. Vector restoration can be used to directly restore the original image, thereby avoiding the generation of the residual image (see [66] and footnote 6). For consistency with the standard hybrid coder structure, we have presented it here as an operation to reconstruct the residual image.

At the decoder, only one restoration operation is performed, regardless of the codebook structure:

$$C_{\text{VR-decoder}} = 2k \tag{5.23}$$

Vector quantization proceeds in the same manner. For standard (non-predictive) VQ, the encoder must calculate the distortion between the residual vector and each reproduction vector in the codebook $A(M)$:

$$C_{\text{VQ-encoder}} = 2M \tag{5.24}$$

If a predictive VQ scheme is used, the prediction must be performed once per vector, not once for each codebook entry as in VR. Therefore, the per-pixel encoder complexity of PVQ is,

$$C_{P\text{VQ-encoder}} = 2M + 2k \tag{5.25}$$

Obviously, for binary tree-structured codebooks, the quantity M in both Eqs. (5.24) and (5.25) is replaced by $2(\log_2 M)$

One of VQ's major benefits is its very low decoder complexity; decoding is performed by a simple look-up of the reproduction vector from the received index. Therefore, $C_{\text{VQ-decoder}} = 0$. Conversely, for predictive VQ, the reconstructed vector is generated by applying the global prediction filter, which results in a complexity equal to the VR decoder, given in Eq. (5.23).

The typical encoding steps for a block in a DCT-based coder are: apply the transform, scalar quantize the transformed coefficients, and run-length encode (RLE) the quantized values. The two-dimensional DCT is a separable transform, which can be performing by a applying series of 1-D DCT's on the rows and columns of the image block. Numerous fast implementations exist; for our simulations, we used the "alternative" method described in [57], which requires 12 multiplications and 32 additions per 1-D DCT. For an 8×8 image block, this amounts to 11 operations-per-pixel. The scalar quantization (simple division) and RLE each require approximately 1 operation-per-pixel, resulting in the encoder complexity:

$$C_{\text{DCT-encoder}} = 13 \tag{5.26}$$

	TSVR	TSVQ	DCT
Encoder	240	40	13
Decoder	10	0	13

Table 5.1: Per-pixel complexity comparison for coding schemes.

Decoding performs the inverse of the encoding operations, in reversed order, namely, run-length decoding, scalar multiplication, and inverse DCT. The complexity of the DCT-based decoder is, therefore, equal to its encoder.

Table 5.1 summarizes these results, where we have assumed that $(\log_2 M) = 10$ (0.15625 bpp) and $k = 5$ (the $\aleph_{4+}$ neighborhood was used for the restoration of each residual image pixel). We note the highly asymmetric nature of both VR and VQ, and that the VR decoder is comparable to that of the DCT-based scheme.

5.5.2 Storage

For vector restoration, the codebook $A(M) = \{(\boldsymbol{H}_i, \boldsymbol{c}_i);\ i = 1, \ldots, M\}$ must be stored at both the encoder and decoder. For 8-bit images, each component of the N-dimensional vector $\boldsymbol{c}_i$ can be represented by one byte. Also, since $\boldsymbol{H}_i$ typically is sparse, we only need to store the k_j non-zero coefficients that constitute $\boldsymbol{h}_{ij}$ for each row of this matrix. While in theory, the coefficients can take any real-value, we can utilize the finite, integer-valued range of pixel intensities to represent these quantities with scaled, fixed-point numbers. This will greatly reduce the storage requirement, along with the complexity of computing the restored pixel values since fixed-point arithmetic can be employed.

Assume that pixel intensities are represented by 8-bits, with range $[0, 255]$, and that each row of the matrix $\boldsymbol{H}$ has k non-zero coefficients. To ensure that the error from using scaled, fixed-point coefficients is less than 0.5, the round-off error, ε, for each matrix coefficient is constrained by,

$$|\varepsilon| < \frac{0.5}{255k} \tag{5.27}$$

Therefore, we require a precision of $\lceil \log_2(510k) \rceil$ bits for the fractional portion of each coefficient, where $\lceil \quad \rceil$ denotes the ceiling function. In our simulations, we have observed that the integer portion of the matrix coefficients falls within the range $(-3, 3)$, which is represented by a 3-bit value.

Using the above analysis, a matrix coefficient can be accurately represented by only two bytes if $k \leq 16$. The total VR encoder and decoder storage requirement, in bytes, for this situation is then,

$$S_{\mathrm{VR}} = MN(2k+1) \tag{5.28}$$

If a tree-structured codebook is utilized, the encoder storage requirement approximately doubles due to the internal nodes of the tree; conversely, only the leaf nodes must be stored at the decoder.

For vector quantization, the codebook only consists of the reproduction vectors, $\boldsymbol{c}_i$, which results in identical encoder and decoder storage requirements of:

$$S_{\mathrm{VQ}} = MN \tag{5.29}$$

If predictive VQ is utilized, the global prediction filter also must be stored:

$$S_{\mathrm{PVQ}} = MN + N(2k+1) \tag{5.30}$$

Identical to the VR case, if tree-structured codebooks are utilized, the respective encoder storage requirements for both VQ implementations are approximately doubled.

In common DCT-based residual coder schemes, a single quantization matrix is used for the uniform scalar quantization of the transformed image block coefficients; rate-control is achieved by scaling this matrix according to buffer-fullness. Regardless of whether the quantization matrix is fixed or allowed to vary, only a single integer value must be stored for each of the 64 coefficients. The only other storage requirement for these schemes is the Huffman table for the run-length encoded coefficients. The storage of both the quantization matrix and the Huffman table are insignificant when compared to that required for the VR and VQ codebooks; therefore, we set $S_{\mathrm{DCT}} = 0$.

	TSVR	TSVQ	DCT
Encoder	1.44	0.13	0
Decoder	0.72	0.07	0

Table 5.2: Storage requirement comparison for coding schemes (in megabytes).

Table 5.2 compares the storage requirements of these three schemes, were we have again assumed that $(\log_2 M) = 10$, $N = 64$, and $k = 5$. While the TSVR technique requires substantially more memory than either TSVQ or DCT scheme, we note that it is on the order of that required to store a single 640×480 pixel image frame.

5.6 Experimental Results

In this section, we compare the rate-distortion performance for the vector restoration, vector quantization, and DCT-based residual coding schemes on a representative sampling of the multi-view signals described Section 2.4. For all techniques, the predicted images were generated using a full-search, block-based technique, with 16×16 pixel blocks and half-pixel accuracy. The first frame in each sequence was intraframe encoded, and the remaining frames were predictively coded from a previously decoded reference frame. The optimal (minimum-occlusion) reference frame was always used for the prediction of frames within the dependent view; View 0 was specified as the independent view for all sequences.

To perform a direct comparison with current standards, which perform DCT/scalar-quantization for the coding of the residual image [1, 2, 3, 4], 8×8 image blocks were used in the vector coding techniques. These blocks were obtained by sectioning the luminance and two chrominance components of the residual image into nonoverlapping blocks. While this block size is larger than that typically used in vector quantization implementations for image sequence coding [19, 39, 58], we justify this selection with our requirement of low bit rate coding in the region of less than 0.25 bits-per-pixel, and the consistency with the macroblock/block partitioning of image frames in the MPEG and H.26x standards.

For vector restoration, we experimented with using the predicted image neighborhoods of $\aleph_{0+}$, $\aleph_{4+}$, and $\aleph_{8+}$ as shown in Fig. 5.3; therefore, the predicted image block corresponding to the 8×8 residual image block were of dimension 8×8, 10×10, and 10×10, respectively. At image borders, off-image pixels in the neighborhood were mapped from the symmetric location within the image, about the border.

Our restriction to these three neighborhoods is based on the accuracy of the displacement vector estimation stage and the trade-off between rate-distortion performance and complexity/storage costs. From our analysis of the practical constraints of the prediction process (Section 5.2), we assume that the majority of the distortions (degradations) in the prediction are due to: the non-stationarity of image intensities, the finite resolution of the displacement vector estimates, and the occurrence of displacement discontinuities within the predicted image block. The first two causes of distortion are handled by the small, local regions that we utilize; a much larger neighborhood would be required to compensate for the last cause. Although incremental gains may be possible from using such a region, we feel that these gains do not justify the substantial complexity and storage increases associated with larger-sized regions.

The training sequences used for the design of the vector restoration and vector quantization codebooks[11] were obtained from the coding of sequences using a modified MPEG encoder. A particular block in the residual image, and the corresponding block in the predicted image, were included in the training set if the residual block was encoded in the DCT-based simulation; i.e., if there existed any non-zero quantized transform coefficients. Codebook design was performed in an open-loop structure: the residual/predicted image pairs were generated only once by the DCT-based coder; full-rate codebooks then were designed on this data, which was not re-generated after each iteration of the Generalized Lloyd Algorithm based on a new codebook.

In keeping with our desire for acceptable encoder complexity, we only considered tree-structured variants of the vector restoration and vector quantization schemes. In particular, optimal length-pruning of moderate rate codebooks was performed to generate variable rate codebooks with either a specified average rate or average distortion.

Due to the inherent feedback of the hybrid coder structure, both VR and VQ schemes contained a second coder stage, which performed scalar quantization on residual image blocks that were poorly reconstructed from the first, vector coding stage. Finally, a residual vector (block) in all schemes was coded only if its distortion was greater than a specified threshold.

The first results that we will examine were obtained from the *Manege* sequence. Residual/predicted image vector pairs from the first 25 frames from each view were used as the training set, which contained over 100000 vector pairs. Balanced, binary tree-structured, VR (TSVR) and VQ

11. Separate codebooks were used for the luminance and chrominance image components.

(TSVQ) codebooks were designed to level 10 (i.e., 10 bits-per-vector) using this training set. Three separate VR codebooks were generated for the neighborhoods $\aleph_{0+}$, $\aleph_{4+}$, and $\aleph_{8+}$. The PSNR versus average rate results for the for these codebooks on the training set are shown in Fig 5.5a, where the TSVR_0+ denotes neighborhood $\aleph_{0+}$, and so on.

As expected, VR with neighborhood $\aleph_{0+}$ achieves only incremental improvement over vector quantization. Conversely, a gain of over 1.3 dB, at 0.15625 bpp, is realized by using TSVR_4+ over VQ. We also observe that TSVR_4+ and TSVR_8+ have almost identical performance for rates less than 0.125 bpp, after which the performance curves separate. We speculate that the gain achieved by TSVR_8+ for rates over 0.125 bpp are due to over-training of the codebook.[12]

These codebooks were then pruned to various average rates using the generalized BFOS algorithm [16]. The performance plots for the pruned TSVR (PTSVR) and pruned TSVQ (PTSVQ) codebooks are shown in Fig. 5.5b. Again, we note that using the $\aleph_{4+}$ neighborhood provides the best compromise between rate-distortion performance and complexity/storage costs. For the remainder of the experimental results, we will only consider this neighborhood.

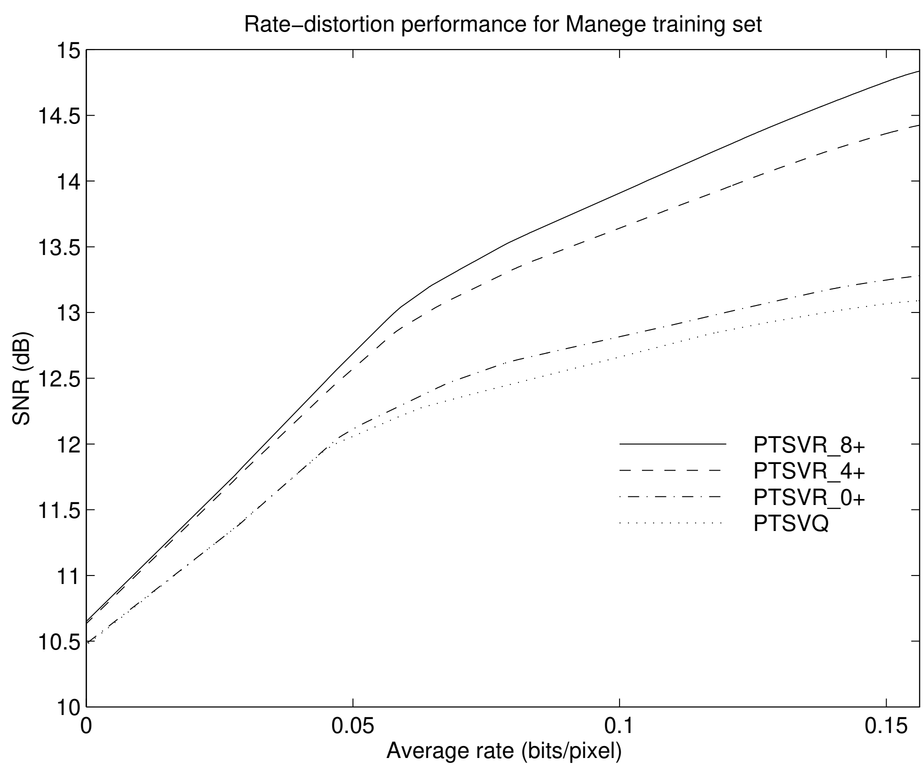
To achieve lower distortion, the TSVQ and TSVR codebooks were extended to levels 13 and 11, respectively, Both were then pruned to an average mean-squared-error distortion of 120. The resulting PTSVR codebook had a rate of approximately 0.09375 bbp, and the PTSVQ codebook 0.15625 bpp. These codebooks were then used to encode the multi-view sequence. The per-frame PSNR and bits required to encode the luminance component for View 0 are shown in Figure 5.6 (average values in parentheses). We have included the image frames used in the training set for two reasons: 1) since the training set was generated using a DCT-based coder and the codebooks were generated in an open-loop structure, its is unlikely that the same residual/predicted vector pairs will occur during the coding of the sequence, and 2) it is useful to compare the performance for image frames within and outside the training set to evaluate the “diversity” of the training vectors.

Vector restoration is able to achieve slightly better PSNR performance while requiring 17% fewer bits to encode the luminance component. No appreciable reduction in this gain is observed between the training (frame numbers 1-25) and the test (frame numbers 26-47) image frames. This is particularly impressive due to the large amount of complex motion in this sequence. In fact, the average predicted image PSNR was approximately 24.6 dB for both schemes.

---

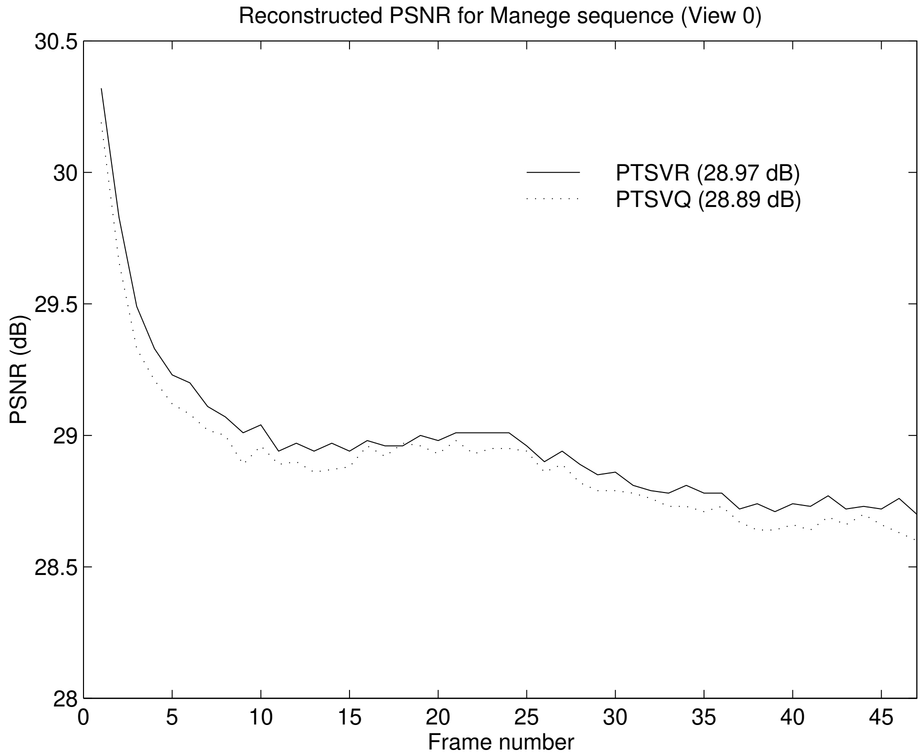
12. While we did not compare VR with a PVQ implementation, we feel that the gain over the residual VQ analyzed here would be constant for all bit rates, and equal to the “zero-rate” gain of VR over VR, i.e., approximately 0.1 dB for $\aleph_{4+}$ and $\aleph_{8+}$, and no gain for $\aleph_{o+}$.

a.

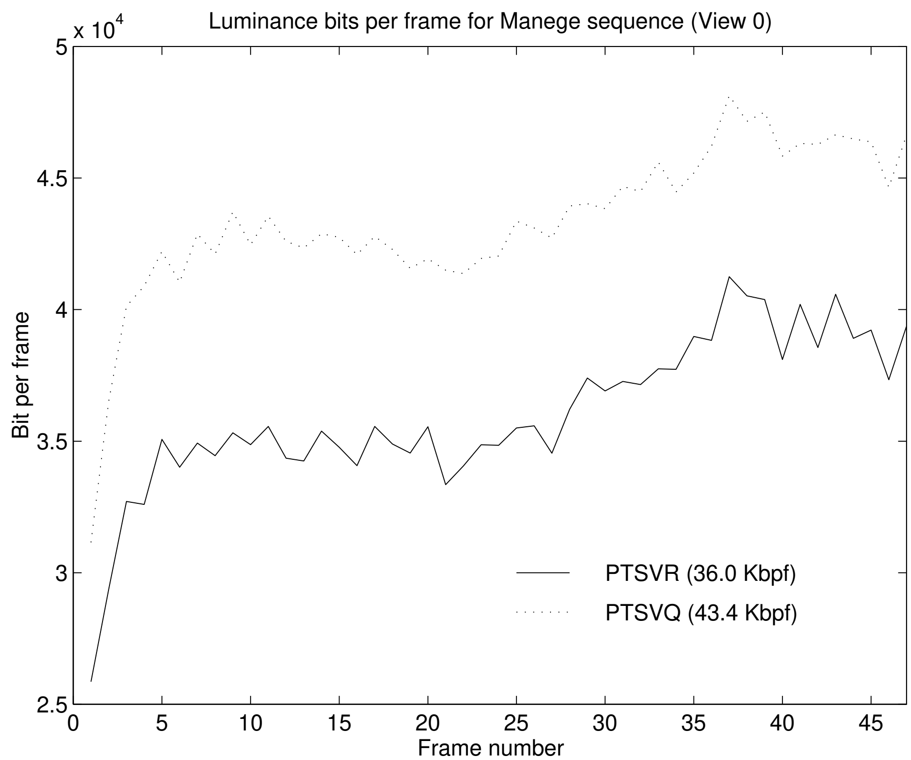


b.

Figure 5.5: PSNR vs. average rate for *Manege* training set, a) balanced, binary trees, b) length-pruned trees.



a.



b.

Figure 5.6: Per-frame performance for View 0 of *Manege* sequence, a) PSNR, b) bits used to encode the luminance image component.

The same steps were performed for the coding of the *Piano* sequence; however, this time, we were interested in achieving a fixed-rate comparison between the VR, VQ, and DCT-based schemes. Balanced, binary tree codebooks were designed to level 10 and then pruned back to level 8, for both VR and VQ, with respective average mean-squared-error distortions of 120 and 168.

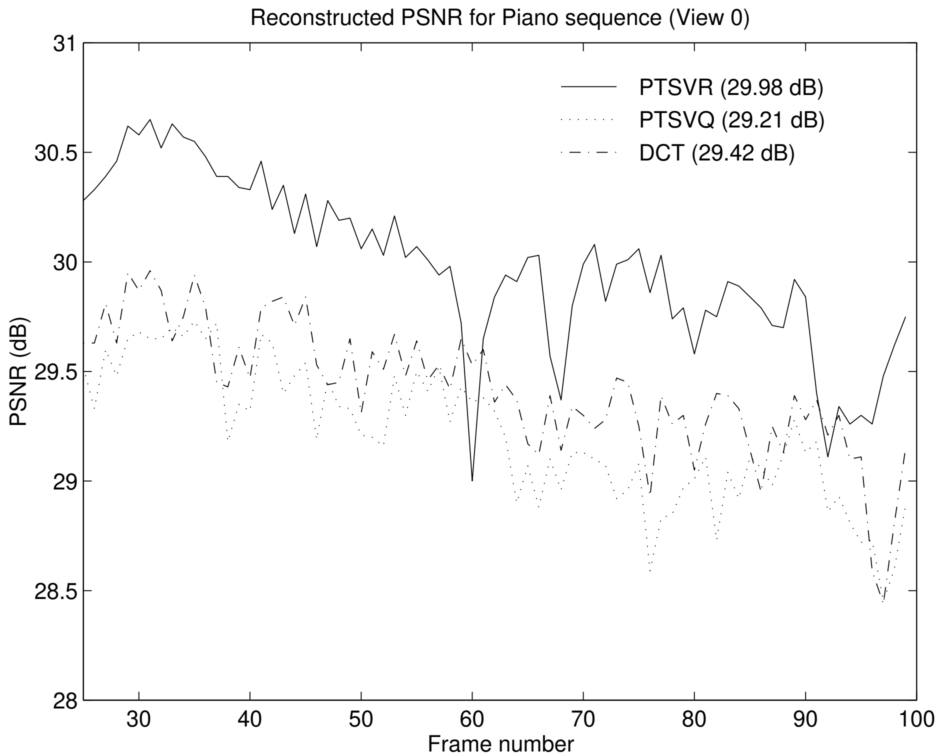
The rate-distortion performance for frames outside of the training set for the two views are shown in Figs. 5.7 and 5.8. Here, VR yields an average PSNR improvement of approximately 0.8 dB over VQ and 0.6 dB over the DCT-based technique. While these average values are not overly impressive, the gain realized by VR was over 1.3 dB for select frames. Also, only approximately one-third of all residual image blocks were coded in the three schemes; therefore, the performance gain of VR is likely to be considerably higher for these blocks than the overall PSNR values indicate.

From our complexity analysis (Section 5.5.1), an average level 8 TSVR encoder using $\aleph_{4+}$ should require 6 times as many operations compared to a comparable TSVQ encoder, and approximately 12 times as many compared to the DCT scheme; however, in our simulations, these values were effectively halved, i.e., the amount of time required to encode a frame using VR was respectively 3 and 6 as much as that required for either VQ or DCT. This deviation from our analysis is likely due to the fact that the entire distortion function need be computed for each codebook entry – if the distortion for part of the vector is greater than the previous minimum, the calculation can be aborted.

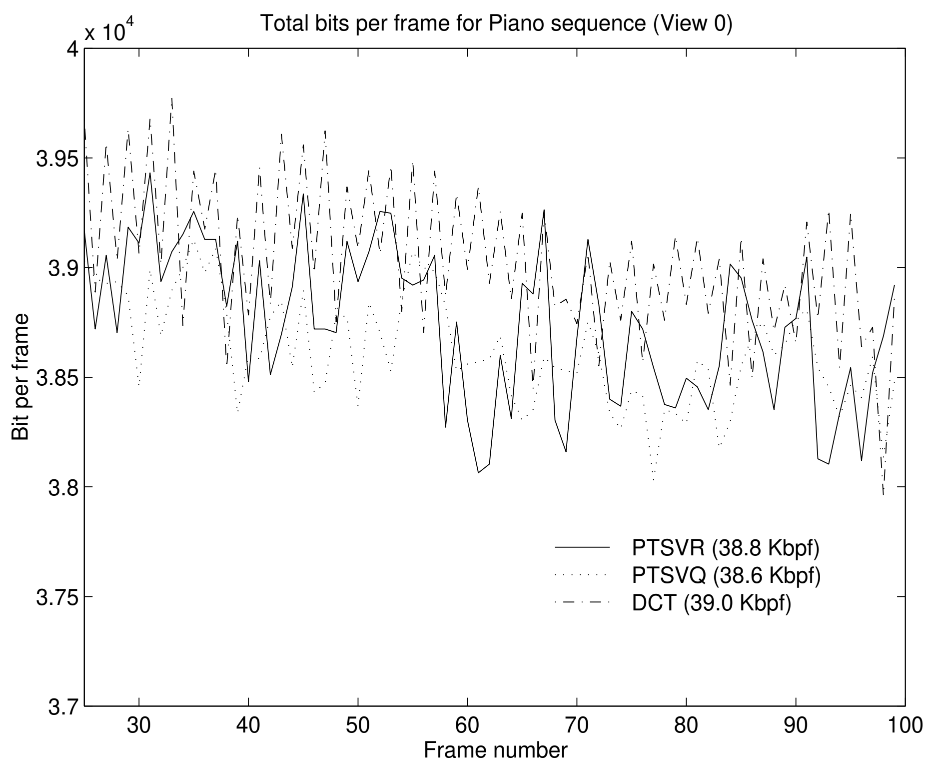
The final results that we will examine were obtained for the two-view, *Finish Line* sequence, where the codebooks used were those generated for the *Manege* sequence. These two sequences possess no common frames or scene objects. The rate-distortion comparison between the PTSVR and PTSVQ implementations is illustrated in Figs. 5.9 and 5.10.

These results are similar to those obtained for the *Manege* sequence, namely, that vector restoration requires approximately 14% fewer bits per frame to encode than vector quantization with a comparable average distortion. We note that the total bits-per-frame values include all prediction and header descriptions, which were approximately equal for both coders. Broken down into the individual luminance and chrominance components of the residual image, the bits-per-frame quantities were: VR: 11999 (Y) and 336 (C); VQ: 14724 (Y) and 926 (C).

This sequence also was coded using the DCT-based scheme. The average PSNR results for this technique were approximately equal to that of VR and VQ; however, it required approximately 10% fewer total bits per frame than VR. From an analysis of the bit allocation, we conclude that this improvement is due to the fact that the VR and VQ implementations contained a second

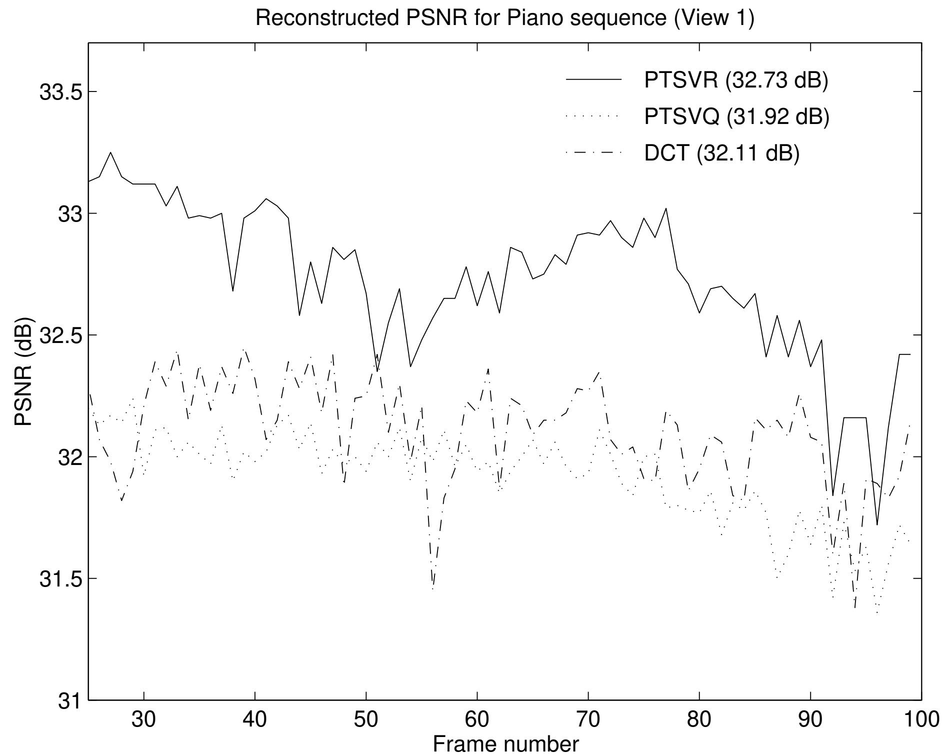


a.

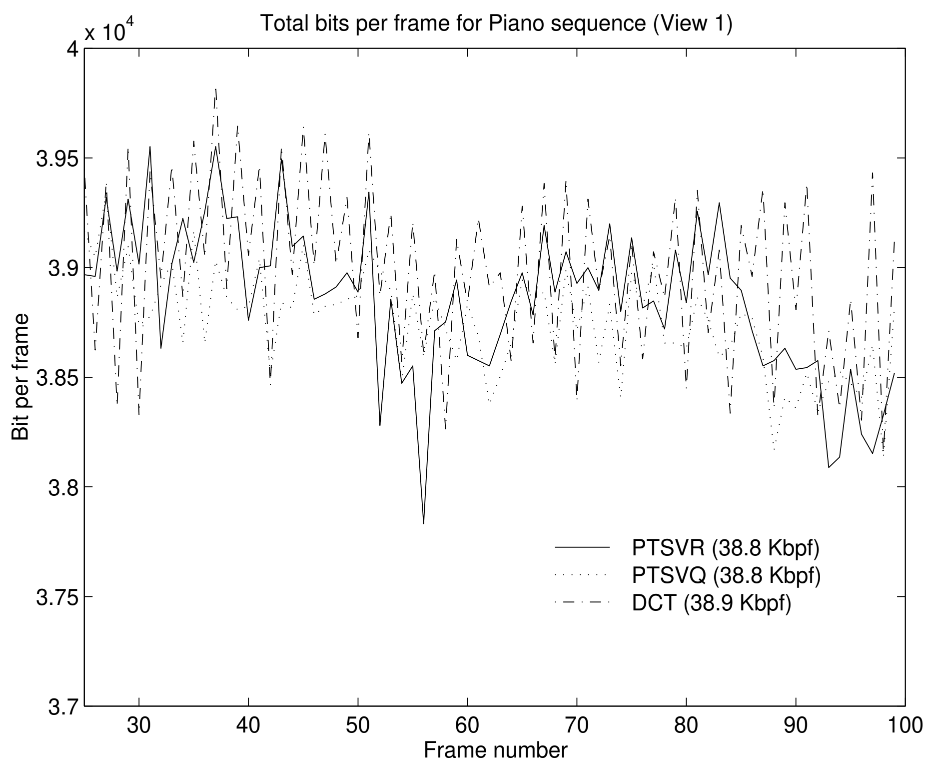


b.

Figure 5.7: Rate-distortion comparison for View 0 of *Piano* sequence, a) PSNR, b) total coded bits per frame.

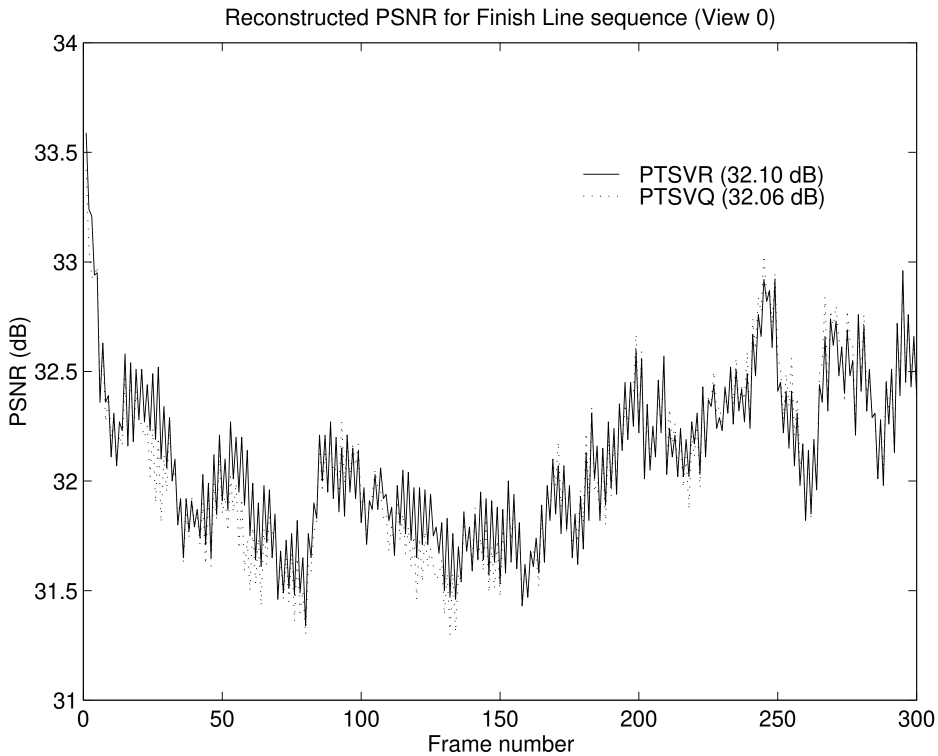


a.

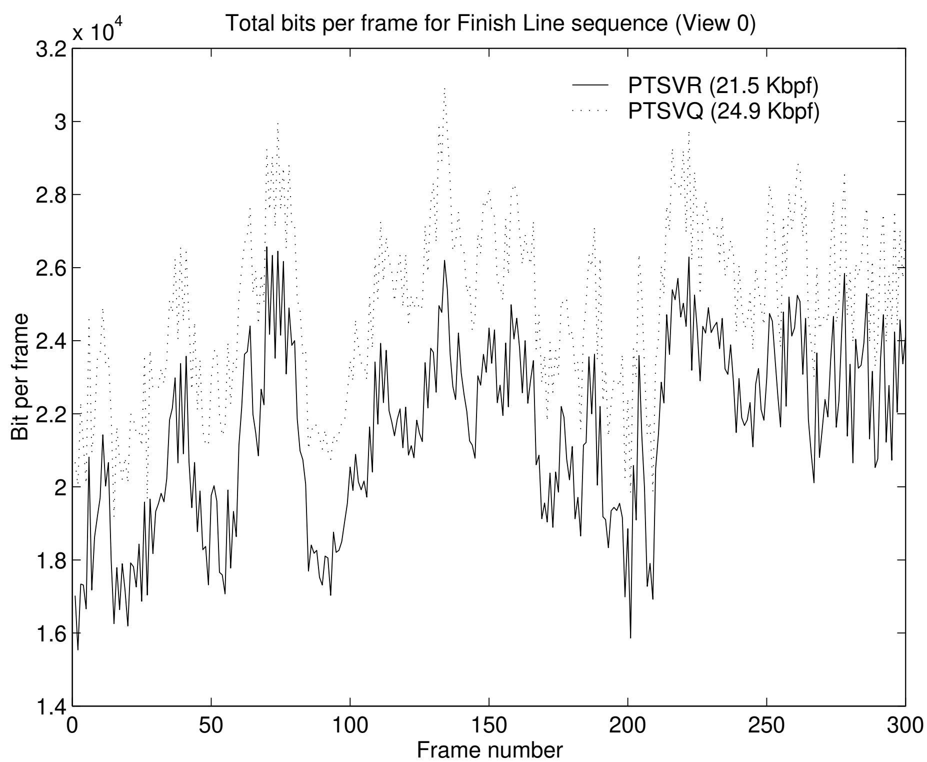


b.

Figure 5.8: Rate-distortion comparison for View 1 of *Piano* sequence, a) PSNR, b) total coded bits per frame.

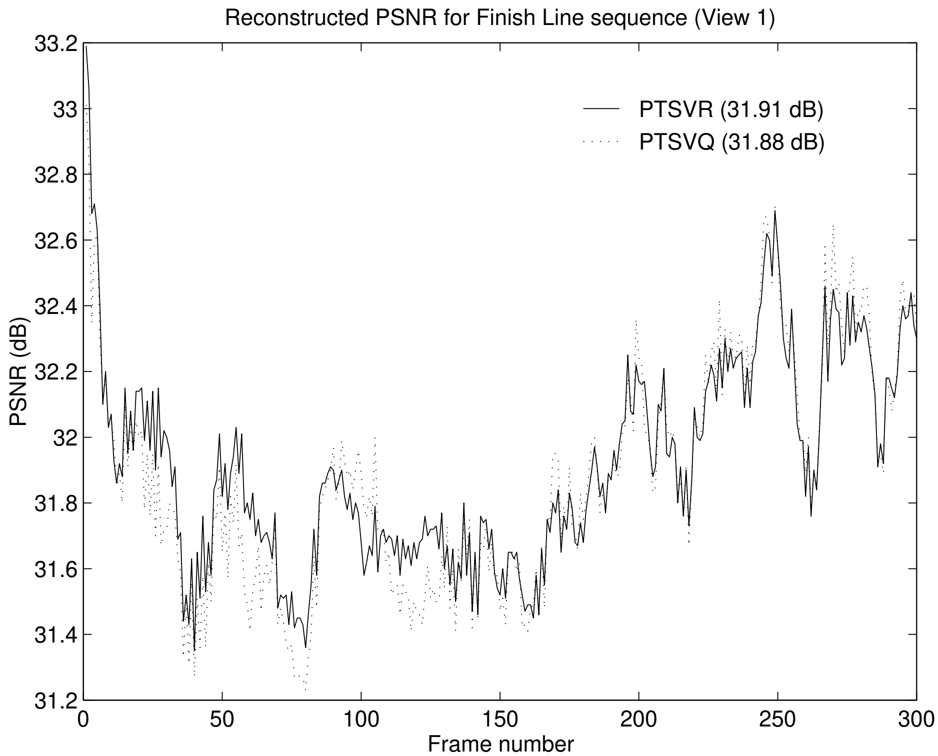


a.

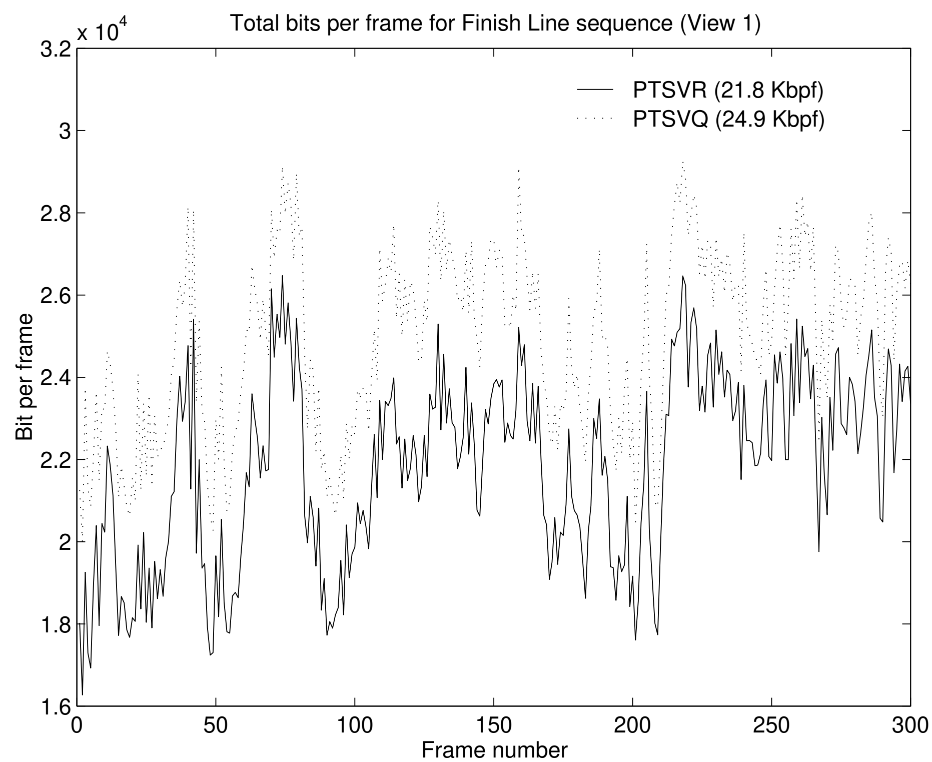


b.

Figure 5.9: Rate-distortion comparison for View 0 of *Finish Line* sequence, a) PSNR, b) total coded bits per frame.



a.



b.

Figure 5.10: Rate-distortion comparison for View 1 of *Finish Line* sequence, a) PSNR, b) total coded bits per frame.

coder stage to handle poorly reconstructed blocks. While this stage was used relatively infrequently (approximately 10% of the image blocks were encoded with this stage), a flag had to be transmitted for each block indicating whether this stage was utilized. Examining only the average bit counts for the luminance and chrominance components of the DCT-coder, 13046 (Y) and 232 (C), we notice that VR performs slightly better. Therefore, it would be advantageous to use a more efficient method of specifying whether the second stage was used, possibly by entropy-encoding these flags for an entire macroblock.

One final observation on this sequence is that we might have been overly optimistic, in terms of using a codebook trained on only one, highly-unrelated sequence; nonetheless, vector restoration still outperformed vector quantization. We have included these results here to demonstrate the problems associated with codebook design.

5.7 Summary

In this chapter, we have presented the novel class of low bit rate, restoration-based residual coders. The basic structure and operation of this technique resembles that of a predictive vector quantizer. However, instead of utilizing a fixed, or slowly varying prediction filter, the codebook contains a set of optimal affine transforms that are applied to the predicted image, in a piecewise manner, to *restore* the residual. Each affine transform essentially represents a coupled linear predictor and quantizer, which are jointly transmitted with one index.

The two main points of this work are that: 1) the non-stationarity of pixel intensities and the practical constraints (namely, the rate and complexity costs) of the prediction process cause the predicted image to be sub-optimal, in the sense that the residual image is not a white-process and is instead correlated with the prediction; and 2) the performance of a residual encoder can be improved by "directing" the encoding process based on the prediction.

The experimental results illustrated the gain achievable by exploiting the predicted-residual image correlation in the low bit rate region of less than 0.25 bits per pixel. Vector restoration consistently achieved superior rate-distortion performance over comparable vector quantization implementations. Gains over a Discrete Cosine Transform scheme were also observed, when the restoration codebook was designed from a representative training sequence. Complexity and storage requirements of this technique were presented. Due to the relatively high encoder complexity cost and the incremental improvement obtained by this technique, we feel that it is most applicable for situations were the rate-distortion performance of the system is of primary concern, e.g., our multi-view system to provide the viewer with 2-D simulated motion parallax.

Future work includes the design of vector restoration codebooks using more representative image sequences to evaluate the viability of this technique for the compression of a diverse class of video signals. We speculate that a method to adapt the codebook will alleviate some of the problems associated codebook generation, in particular the codebook design time and over-training. For example, we plan to apply the concepts of “super-codebooks” [106], codebook replenishment [39], and adaptive filter techniques [85] to our vector restoration coder.

Chapter 6

Interpolation from a Noisy Displacement Vector Field

This chapter considers the problem of interpolating intermediate views[1] from two decoded image frames captured by spatially-offset cameras. This work has application in multi-view binocular display systems to provide the viewer with simulated motion parallax and interactive control over the perceived depth range in the imagery. We first discuss the viewpoint interpolation problem and illustrate the simple procedure of interpolating unoccluded regions when the actual displacement information is known. We then present an improved technique that accurately handles occlusions, displacement estimation errors and ambiguously-referenced image regions.

The key contributions of this work are the development of: 1) the *self-synthesis* procedure, which uses the occlusion-displacement discontinuity relationship (Section 3.2.2) to refine an initial displacement vector field and to detect occlusion boundaries, 2) scene assumptions that yield estimated range information for occluded regions, and 3) a method to accurately select the correct displacement for interpolated image regions that are referenced by more than one displacement vector. A viewpoint interpolator has been implemented that incorporates these enhancements and requires only one block-based displacement estimation operation. A representative set of interpolated views are illustrated, and comparisons are made with displacement-compensated interpolation schemes that do not use these enhancements. For the one sequence where the actual intermediate views were available, the average peak signal-to-noise ratio of the interpolated views was approximately 27.5 dB. The interpolated views have been displayed on a multi-view binocular imaging system, with subjectively effective motion parallax and control over the perceived depth range. These gains are obtained without any additional bit rate beyond that required to encode the two reference frames.

---

1. An intermediate view is defined as the image that would be obtained from a virtual camera located between and on a straight-line connecting two cameras in a multi-view signal.

6.1 Viewpoint Interpolation

We are concerned with generating subjectively-pleasing intermediate views from two decoded image frames within a multi-view signal. We require that the given views were sampled at the same time instant, and were captured by cameras offset in only their horizontal and/or vertical position. These restrictions will allow us to make reasonable scene assumptions, which provide for the rectification of displacement estimation errors and the accurate interpolation of occluded and ambiguously-referenced image regions.

After briefly illustrating our applications for the interpolated views, we describe the process of viewpoint interpolation for unoccluded regions. This leads to a discussion on the effects of occlusions and displacement estimation errors on the quality of the interpolated views. We then summarize prior work that has addressed these problems.

6.1.1 Motivation

We have previously discussed the application of providing a viewer with simulated motion parallax in Section 1.1.2. Improved depth perception is achieved by presenting the appropriate image (or pair of images for binocular systems) based on the viewer's location.

Intermediate views also can be used to ensure that stereoscopic imagery is comfortable to view. Discomfort is often experienced when viewing stereoscopic images on a two-dimensional display device. The breakdown of the accommodation/convergence relationship has been widely reported as a cause of this discomfort [55, 62, 65, 74, 103]. An additional source of eye strain is related to the viewer's ability to fuse binocular information [97].

To achieve geometrically correct stereo-vision, in the sense that the viewer sees what would be seen by the naked-eye, the cameras capturing the binocular image pair must be separated by the viewer's inter-ocular separation [33]. However, this quantity is viewer dependent and individuals often prefer varying degrees of depth perception based on individual stereoscopic viewing ability and the range of depth present in the scene [97]. A greater sense of depth is provided by a relatively large inter-camera separation, but the larger the separation the more difficulty marginal viewers have in fusing the images. If a continuum of views between two extreme viewpoints are available, a viewer can dynamically select the inter-camera separation for comfort and preferred sense of depth in a manner similar to the adjustments of brightness and contrast found on most display devices [63, 67, 97]. Decreasing the camera separation also reduces the breakdown between the accommodation/convergence relationship.

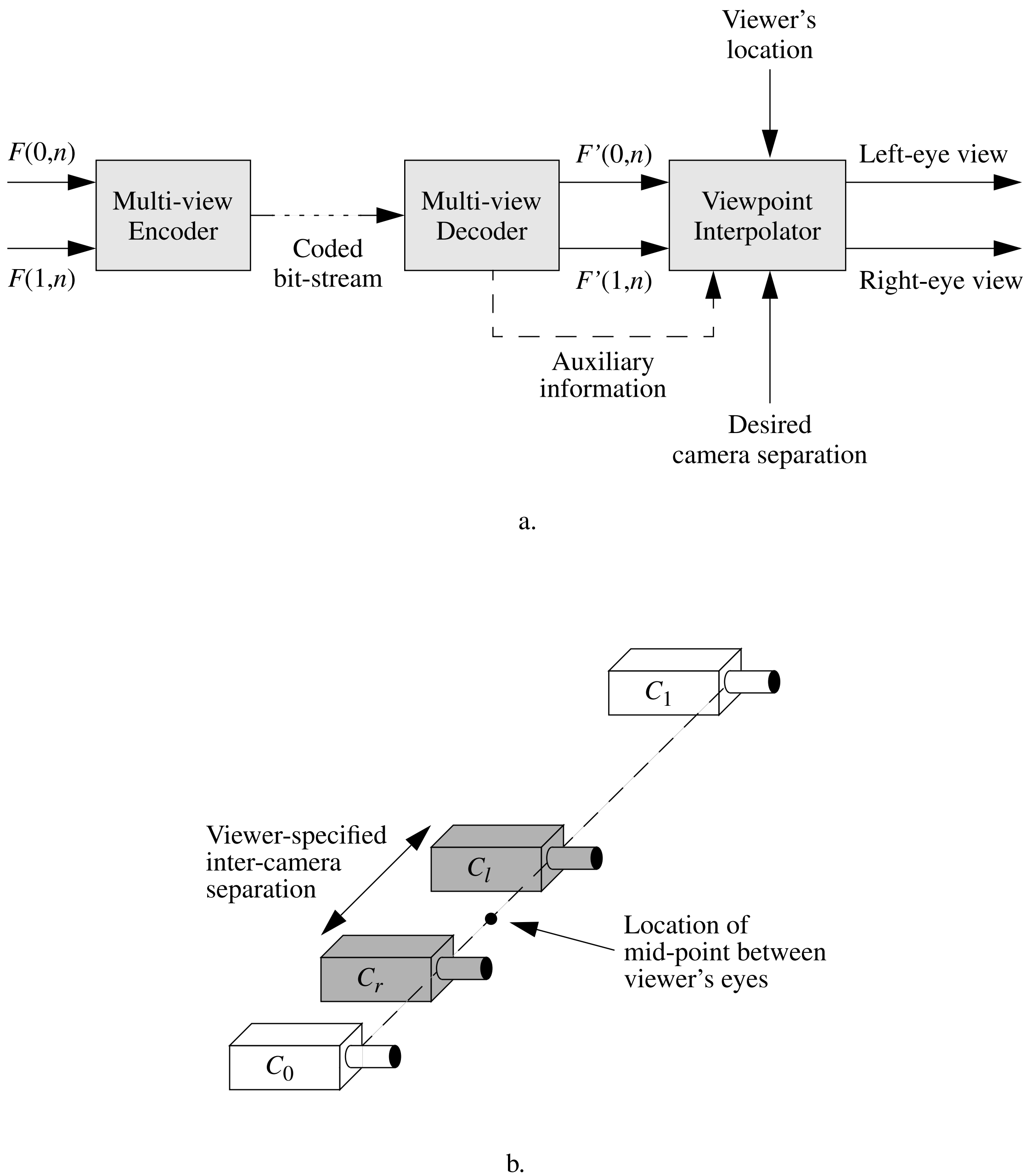


Figure 6.1: Multi-view system for simulated motion parallax and viewer-specified inter-camera separation, (a) functional block diagram, (b) virtual camera positions of desired intermediate views.

Since an infinitesimal camera spacing is required for the motion parallax and viewer-specified inter-camera separation applications, it is impractical to capture and encode all of these views. Instead, we wish to compress the two extreme views and synthesize the desired intermediate views at the decoder. Figure 6.1 illustrates such a system for a two view signal. The viewpoint interpolator takes as its input the two reconstructed views, the parameters that describe the location of the desired intermediate views, and any auxiliary information extracted by the decoder that is useful to the interpolation process. The virtual camera positions of the interpolated views lie between the viewpoints of the extreme views, and are depicted by the shaded cameras for the left-eye (C_l) and right-eye (C_r).

6.1.2 Unoccluded Regions

We assume that the desired intermediate viewpoint is located between Views 0 and 1 of a decoded multi-view signal. Applying our camera geometry constraint, the differential viewpoint parameter matrix of View 1 with respect to View 0 is:

$$\Delta\Phi_1 = \begin{bmatrix} \Delta c_x & 0 \\ \Delta c_y & 0 \\ 0 & 0 \end{bmatrix} \tag{6.1}$$

The intermediate view is parameterized by its relative position between the given views by β, where $0 \le \beta \le 1$ and,

$$\Delta\Phi_\beta = \beta \cdot \Delta\Phi_1 \tag{6.2}$$

Viewpoint interpolation is related to the range of objects in the scene. Since we have restricted the cameras to have coplanar image sensors, parallel camera axes and coincident roll-vectors, only translational displacement of objects between views is possible. As such, unoccluded regions are interpolated by linearly scaling the displacement field relating corresponding points in the reference views by β. For now, we assume that the locations of occluded regions are known and that the actual displacement field relating corresponding points in View 0 from View 1, i.e., $\boldsymbol{d}_{0 \leftarrow 1}(u, v)$, is provided. The vector field that describes the displacement of unoccluded pixels from View 0 to the intermediate view is obtained from,

$$\boldsymbol{d}_{\beta \leftarrow 0}(u - \beta d_{u_{0 \leftarrow 1}}(u, v), v - \beta d_{v_{0 \leftarrow 1}}(u, v)) = -\beta \boldsymbol{d}_{0 \leftarrow 1}(u, v) \tag{6.3}$$

If $\beta = 0$, the location of the intermediate view obviously coincides with that of View 0, and all displacement vectors are zero. Conversely, if $\beta = 1$, the desired view is merely View 1 and Eq. (6.3) reduces to Eq. (3.9). The symmetrical relationship that provides the interpolated displacement field from View 1 is obtained by substituting the appropriate displacement vector components and $(1 - \beta)$ for β in the proceeding equation.

Even though a one-to-one relationship exists between bidirectional displacement vectors relating Views 0 and 1, it is possible for two distinct vectors to satisfy Eq. (6.3), i.e., a many-to-one mapping of displacement vectors may exist. This situation occurs when an object lies between the camera pair and another object, and the magnitude of the difference between the displacements of the two objects is greater than the dimension of the foreground object along an epipolar line. Since the foreground object will occlude the background object, the interpolated displacement vector for this region should be set to the scaled displacement vector of the foreground object. The range ordering of objects is a function of the relative displacement values of the foreground and background objects and the camera geometry.

Once the interpolated displacement field for unoccluded pixels has been calculated, the intensity values for these pixels are predicted equivalently from either given view using the standard displacement relationship (see Section 3.1.3):

$$\begin{aligned} \hat{I}(u, v, \beta) &= I(u - d_{u_{\beta \leftarrow 0}}(u, v), v - d_{v_{\beta \leftarrow 0}}(u, v), 0) \\ &= I(u - d_{u_{\beta \leftarrow 1}}(u, v), v - d_{v_{\beta \leftarrow 1}}(u, v), 1) \end{aligned} \tag{6.4}$$

We illustrate the process of viewpoint interpolation for unoccluded regions in Fig. 6.2. This figure depicts the familiar one-dimensional displacement field along an epipolar line. The displacement of a pixel in the interpolated view is equal to the appropriately-scaled displacement between corresponding points in Views 0 and 1 that intersects the desired pixel. The intermediate pixel is mapped from the pixel at either end-point of the intersecting vector.

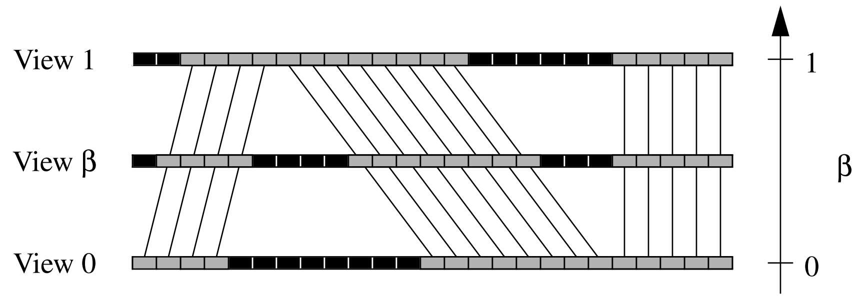


Figure 6.2: Interpolated displacement vector field along epipolar line.

If the camera geometry is known and if we are provided with the true displacement vector field, unoccluded pixels are interpolated simply from Eqs. (6.3) and (6.4). However, so far, we have neglected two important considerations: 1) the displacement vector field is unknown and must be estimated, and 2) a displacement vector for each pixel in the interpolated view cannot be generated due to occlusions in the given views. We next describe the effects of displacement estimation errors and occlusions on the subjective quality of the interpolated view.

6.1.3 Effects of Displacement Estimation Errors and Occlusions

The first step in the displacement-compensated interpolation process is the estimation of the displacement vector field relating corresponding points between the two given views. Regardless of the method used to compute the displacement field, errors are likely to occur. We classify both inaccurate estimates for unoccluded regions and meaningless displacement values for occluded regions as errors. Estimation errors propagate to the interpolated displacement field, and result in pixels that are interpolated incorrectly.

Our experience has shown that an erroneous displacement vector will not result in a visually-discernible artifact in the interpolated image provided that the error occurs in an image region with relatively uniform intensity, or that the individual components of the estimated vector are within one pixel of the actual values. Therefore, exact range information of the scene is not required to yield subjectively-pleasing interpolated views.[2]

We have also observed that the most objectionable artifacts occur when: 1) an estimation error occurs in a highly textured region, 2) the magnitude of the error is greater than two pixels, 3) an

2. This represents a significant computational reduction compared with applications that require a solution to the ill-posed problem of generating an exact, dense range map of the scene [90].

inaccurate boundary between unoccluded regions with differing displacements is estimated, or 4) the meaningless displacement estimates for occlusions are used in the calculation of the interpolated field. These errors not only adversely affect the interpolation of individual pixels, they may also destroy the *displacement continuity* of objects, which occurs when poorly-interpolated views are used in the motion parallax and/or viewer-specified inter-camera separation applications. As consecutively-spaced interpolated viewpoints are presented to the viewer, these serious errors will cause portions of objects at identical depth planes to appear to move with varying displacements. This situation is particularly detrimental to the subjective quality of the interpolated imagery, and may negate any benefits provided by synthesizing intermediate views of the scene.

An object's displacement continuity is often violated when two estimated displacement vectors intersect at a pixel location in the interpolated field. For example, assume that one of these estimates is correct and the other is false. Also assume that the displacement of the false estimate indicates that this pixel lies in front of the other pixel. Even though the true displacement for this pixel was estimated correctly, one would likely select the incorrect displacement vector due to the range ordering of actual unoccluded regions. If proper displacement vectors are used for the interpolation of neighboring pixels, this pixel will appear disjunct from the adjacent image region.

We refer to interpolated image regions that are intersected by more than one displacement estimate as being *ambiguously-referenced*. Due to the non-ideal displacement estimation process, the selection between which estimate to use for these regions should not depend solely on apparent range ordering.

Even if it is possible to generate an error-free DVF, *holes*, or regions that do not have a valid displacement vector, will be present in the interpolated displacement field. Holes are due to occlusions in the given reference frames, or may result from displacement estimation errors. Since we have no direct knowledge of the displacement for these regions, we cannot use Eq. (6.4) to interpolate these pixels. We are also hampered by the fact that the locations of the occluded regions are unknown; we cannot differentiate whether the hole is due to an actual occlusion or an estimation error. The un-interpolated regions typically occur in clusters, which may contain a significant number of pixels;[3] therefore, simple region filling procedures will not yield satisfactory results.

An actual one-dimensional displacement field of a sample scene and its corresponding estimated field are depicted in Fig. 6.3. The estimated field contains numerous occlusions and an ambiguously-referenced region. Holes will exist in the intermediate image at pixels that lie within

3. For reference frames with moderate amounts of occlusion, holes typically occupy 5-10% of the pixels in the interpolated view.

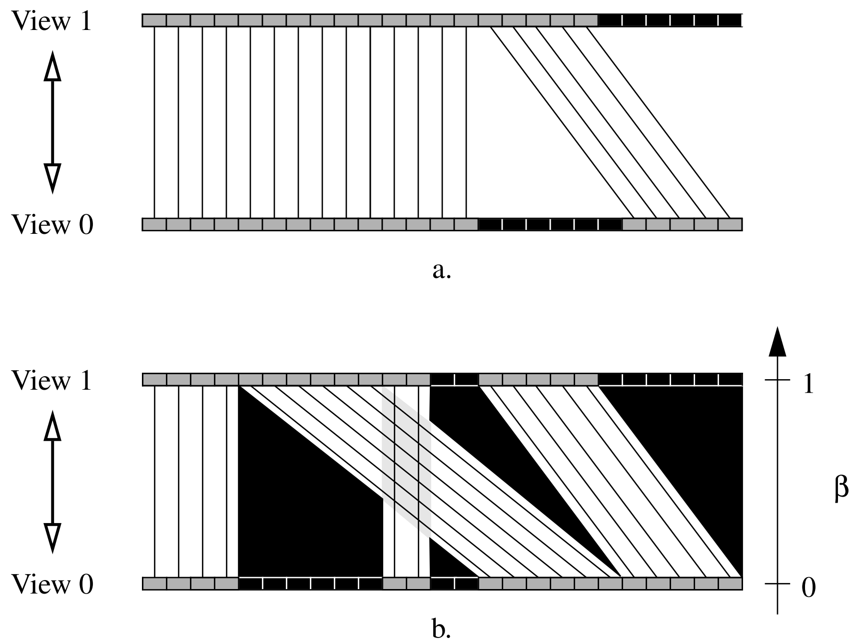


Figure 6.3: (a) Actual displacement field, (b) estimated field with occluded (blackened) and ambiguously-referenced (lightly-shaded) image regions.

the blackened region; pixels that lie within the lightly-shaded region are intersected by two displacement vectors. It is unclear how these regions should be interpolated. These "difficult-to-interpolate" image regions are particularly troublesome since we have explicitly assumed that the actual intermediate view is not available and, thus, residual coding of poorly interpolated regions is not possible.

In summary, to improve the subjective quality of the interpolated image, we wish to: 1) reduce displacement estimation errors, 2) select the proper displacement vector for ambiguously-referenced regions, 3) accurately localize the boundaries between occluded and unoccluded regions, and 4) infer range information for occluded regions. If these goals can be obtained, accurate and robust interpolation of the complete intermediate view is possible. We next describe prior work that has addressed at least some of these issues.

6.2 Prior Work

The majority of work dealing with the interpolation of spatially-offset views has used more than two views of the scene to minimize the effects of occlusions and displacement estimation error.

A common interpolation approach is through the generation of epipolar plane images [30, 40, 47]. We have discussed the various advantages and disadvantages of these non-arbitrary multi-view signals in detail in Section 2.2. Essentially, numerous views of the scene are captured and processed at the encoder to estimate the displacement variation of objects through the set of image frames. Improved displacement estimation is possible due to the repeated observations, and the minute spacing between the views minimizes the size and number of image regions visible in one, and only one, view.

Both EPI-based techniques reported by Katayama, *et. al*, and Hsu, *et. al*, handle holes in the interpolated image by simple region filling with adjacent pixel data. In [40], ambiguously-referenced image regions are addressed through the use of *aspect graphs* [25] to layer scene objects based on range; hidden surface removal then is performed to ensure that foreground objects are mapped in front of background objects. A more cautious approach to interpolating image regions with more than one possible displacement trajectory is taken in [47]. Here, the selection is based on a measure of the reliability of the intersecting displacement estimates; the displacement estimates from regions that have fewer valid trace lines through the EPI are assumed to be more accurate than those that have numerous valid traces.

A slightly different EPI-based approach was described by Fujii and Harashima [30]. This work was presented in the context of both compressing the multiple views and synthesizing virtual viewpoints. A more compact representation of the pixel displacements, compared to trace lines, was obtained using triangular patches of textured regions. The grid nodes of the triangular mesh were specified by minimizing the *Var*-space, which represents the prediction error of a mesh for a particular displacement. The initial algorithm was observed to have difficulty when occlusions were present in the scene; this was addressed through the segmentation of the *Var*-space based on depth.

Skerjanc and Liu presented a technique for synthesizing intermediate views using three cameras positioned at the corner-points of a right-angled isosceles triangle [89]. An object-oriented, feature matching approach was used to analyze the range information of the scene. Displacement correlation between epipolar-lines is exploited through the use of feature points, as opposed to the single-line EPI displacement estimation methods. Even though the additional view reduces the amount of occluded regions and the occurrence of displacement estimation errors, it is unlikely that these problems will be completely eliminated; however, no solution to these problems was provided in their work.

A larger body of research has been devoted to the related problem of interpolating images frames from two temporally-offset images captured by the same camera. Motion-compensated interpolation is used for the applications of frame rate conversion, de-interlacing and frame skipping [13, 48, 83, 93]. Since constant, translational object motion is often assumed, interpolation of unoccluded regions is performed in the same manner as described in Section 6.1.2.

Cafforio, *et. al.*, achieve a highly accurate motion field by using bidirectional motion estimates obtained from a pel-recursive algorithm [13]. To reduce "streaking effects" at object boundaries, a total of eight recursions must be performed and combined to yield the final vector field. They presented an interesting procedure to detect occlusions that uses the one-to-one relationship between the displacement of unoccluded regions.[4] After direct interpolation of unoccluded regions, the authors reported that holes occupied only 2% of the intermediate view, which were then filled using spatial interpolation. However, a relatively small amount of occlusion must have been present between the reference frames to yield such a limited number of holes – an accurate interpolation should be fairly simple for these images. No mention of ambiguously-referenced regions was provided.

Thoma and Bierling explicitly consider the effects of occluded regions in their motion-compensated interpolation algorithm [93]. They combined a hierarchical displacement estimator with a change detector to improve the accuracy of the displacement vector field and to classify occlusions as being covered or uncovered between the two reference frames in the image sequence. Holes were assumed to be due to occlusions of the background; they were assigned zero displacement and mapped from the appropriate frame in which region was visible. The change detection and occlusion mapping steps rely on the assumption of a stationary background. While this assumption may be valid for image sequences, it is not valid for spatially-offset views – only a single-depth plane will exhibit zero displacement.

A motion-compensated interpolation scheme that considers object scaling was presented by Ribas-Corbera and Sklansky [83]. They used a bidirectional optical flow analysis and Cafforio's occlusion detection algorithm to generate the displacement field containing only likely unoccluded regions. Again, holes were filled using a zero displacement mapping from the reference frame that contained the occlusion. Due to the more elaborate, non-translational motion model, ambiguously-referenced regions could be due to both occlusion and scaling; they were handled by selecting the displacement with the greatest slope, i.e., via range ordering. The results of this technique indi-

4. This approach is similar to our independently developed technique for displacement field reversal, described in Section 4.4.1.2.

cated incremental improvement over Cafforio's; yet, the reference frames used in the comparison had relatively modest maximum displacements of 12 pixels and limited occlusion.

While the techniques that use three or more views to interpolate the desired intermediate views often provide impressive results, we feel that the additional views are essentially overkill and unnecessarily increase the camera and encoder complexity. If all image regions within the actual intermediate view are visible within at least one of the extreme viewpoints, we contend that it is possible to interpolate subjectively-pleasing views from only two reference views. Also, many of the assumptions used in the motion-compensated interpolation techniques to handle occluded and ambiguously-referenced image regions are only valid for temporally-offset image frames. We next describe our improved technique for synthesizing intermediate viewpoints from two, spatially-offset views that accurately interpolates these difficult image regions.

6.3 Improved Interpolation

Since we are concerned with viewpoint interpolation within a complete multi-view encoder/decoder system, we assume that the estimated displacement vector field relating corresponding points between the extreme views is provided from the predictive coding of one of these views from the other. This auxiliary information (see Fig. 6.1) is readily extracted from the coded bitstream, which alleviates the need to perform a displacement estimation procedure at the decoder.[5]

To facilitate the application of this work to standardized compression algorithms, we assume that the displacement field was estimated using a fixed-size, block-based technique [1, 2, 3, 4, 60]. While block-based techniques provide an adequate prediction and a compact representation of the displacement field, they are prone to serious estimation errors. Displacement errors are due to: 1) displacement discontinuities within an image block, 2) occluded regions, 3) blocks with uniform intensity, and 4) image noise. From our discussion in Section 6.1.3, we are most interested in rectifying errors due to the former two causes.

We view the initial displacement field as a rough approximation of the actual displacement of objects in the scene. We first preprocess the initial field to eliminate likely false estimates using our set of tools developed in Chapter 4 to detect unoccluded regions, namely, SNR thresholding, field reversal and border elimination. We assume that the resulting bidirectional displacement field con-

5. Obviously, if one of the extreme views was not predictively coded from the other view, the interpolation operation must include the estimation of the displacement vector field.

tains only highly reliable estimates for a majority of the unoccluded pixels in the two reference views.

The preprocessed field will contain numerous holes due to the elimination of displacement vectors that were assumed to be erroneous. While many of these holes correspond to occlusions in the reference views, both correct and incorrect displacement estimates for unoccluded regions might have been eliminated. We would like to estimate displacement vectors for unoccluded regions that have been invalidated by using neighboring, valid estimates. Also, due to the block-based displacement estimation procedure, displacement discontinuities will exist only at image block boundaries; it is unlikely that these arbitrary boundaries represent the true junction between scene objects located at varying depth planes. Therefore, we perform a two-pass, *self-synthesis* operation to fill *improper* holes and refine the location of displacement discontinuities.

The self-synthesis procedure essentially interpolates one of the given views from the other view and modifies the displacement vectors of regions that yield poor reconstruction. It uses the fundamental principle that a displacement discontinuity in one view corresponds to an occlusion in the other view, whose length, along an epipolar-line, is equal to the magnitude of the discontinuity. Holes that do not satisfy this relationship are classified as being *improper*, and are filled using displacement vectors from adjacent regions with valid vectors. The self-synthesis operation also searches for the minimum-distortion location for all displacement discontinuities along an epipolar-line. Adjusting the discontinuity boundaries in a unidirectional field is tantamount to localizing occluded regions in the reversed field.

The first pass fills improper holes and adjusts displacement discontinuities within View 0, and the second pass performs the equivalent operation on View 1. This process effectively rectifies many of the displacement estimation errors due to the block-based technique, and provides a significantly complete field to generate an accurate, interpolated DVF.[6]

After the displacement field has been processed, we calculate the interpolated displacement field from pixels with valid displacement estimates using Eq. (6.3). Even though the processed field has considerably fewer errors than the initial field, we are still wary of violating the displacement continuity of scene objects. Therefore, we handle ambiguously-referenced pixels by combining range ordering with a displacement estimation confidence measure to select the proper vector. If a pixel in the interpolated view is referenced by more than one displacement vector and the predicted block-SNR of one of the vectors is significantly greater than the other vectors, we select this

---

6. We note that the preprocessing and self-synthesis are performed only once per pair of reference views, regardless of the number of intermediate views to be synthesized.

vector for the interpolated field. Otherwise, we select the vector from the object closest to the cameras.

We complete the interpolated field by inferring range information for occluded regions (*proper holes*). We make the reasonable scene assumptions that occluded regions have constant depth, and that this depth is equal to the depth of the *appropriate* unoccluded region adjacent to the occlusion. All occlusions, that do not occur at image borders, are bounded by two unoccluded regions, along an epipolar-line, that correspond to two scene objects at different depth planes. The occlusion results from the covering of one of these objects by the other object. The appropriate unoccluded region is defined as the region that is the farther of the two regions from the camera pair, i.e., the background unoccluded region. We use knowledge of the camera geometry and the relative displacements for the unoccluded regions surrounding an occlusion to calculate which region is behind the other. Occluded pixels in the interpolated field are assigned displacement vectors equal to that of the background region, and these pixels are marked to indicate in which view the occlusion was visible.

Finally, we perform the actual interpolation from the complete, interpolated displacement field. Pixels that were referenced by one or more valid vectors are assumed to be unoccluded, and are predicted from a weighted average of the corresponding pixels in the two decoded reference views:

$$\hat{I}(u, v, \beta) = (1-\beta) \cdot \bar{I}(\boldsymbol{p}_0, 0) + \beta \cdot \bar{I}(\boldsymbol{p}_1, 1) \tag{6.5}$$

$$\boldsymbol{p}_0 = \begin{bmatrix} u - \hat{d}_{u_{\beta \leftarrow 0}}(u, v) \\ v - \hat{d}_{v_{\beta \leftarrow 0}}(u, v) \end{bmatrix} \text{ and } \boldsymbol{p}_1 = \begin{bmatrix} u - \hat{d}_{u_{\beta \leftarrow 1}}(u, v) \\ v - \hat{d}_{v_{\beta \leftarrow 1}}(u, v) \end{bmatrix} \tag{6.6}$$

where $\boldsymbol{p}_0$ and $\boldsymbol{p}_1$ respectively denote the pixel locations in Views 0 and 1. The intensities for fractional pixel locations are obtained through bilinear interpolation of the reference view. Averaging pixels from both views for unoccluded regions reduces the effect of high frequency noise components in the interpolation process, and it “blends” the intensity variations between the two exterior views.

Proper holes in the interpolated field, that were filled by inferring range information for occluded regions, are predicted directly from the corresponding pixel in one of the reference views, based on the visibility of the occlusion:

$$\hat{I}(u, v, \beta) = \begin{cases} \bar{I}(\boldsymbol{p}_0, 0), & \text{if visible in View 0} \\ \bar{I}(\boldsymbol{p}_1, 1), & \text{if visible in View 1} \end{cases} \tag{6.7}$$

A flow-chart depicting the entire viewpoint interpolator is provided in Fig. 6.4.

The following three sections provide a detailed description of our novel enhancements to the basic interpolation operation that, respectively: 1) refine an initial, error prone displacement field, 2) accurately select the proper displacement estimate for ambiguously-referenced image regions, and 3) fill holes in the interpolated field by inferring range information for occluded regions. We then present a cost analysis of our improved viewpoint interpolator, and present interpolation results for numerous binocular sequences.

6.3.1 Field Refinement via Self-Synthesis

The self-synthesis procedure adjusts the location of displacement discontinuities and fills improper holes in the preprocessed field by evaluating the performance of interpolating each of the given views from the other view. In this section, we present the procedure for refining the forward displacement field (self-synthesis of View 0 from View 1); the corresponding operation in the opposite direction is obtained simply through the substitution of appropriate quantities in the following discussion.

Since the displacement of corresponding points within spatially-offset views is constrained to one-dimension, the field refinement is performed along individual epipolar-lines. For simplicity, we assume that the reference views were obtained from cameras that were offset in only their horizontal positions:

$$\Delta\Phi_1 = \begin{bmatrix} \Delta c_x & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{6.8}$$

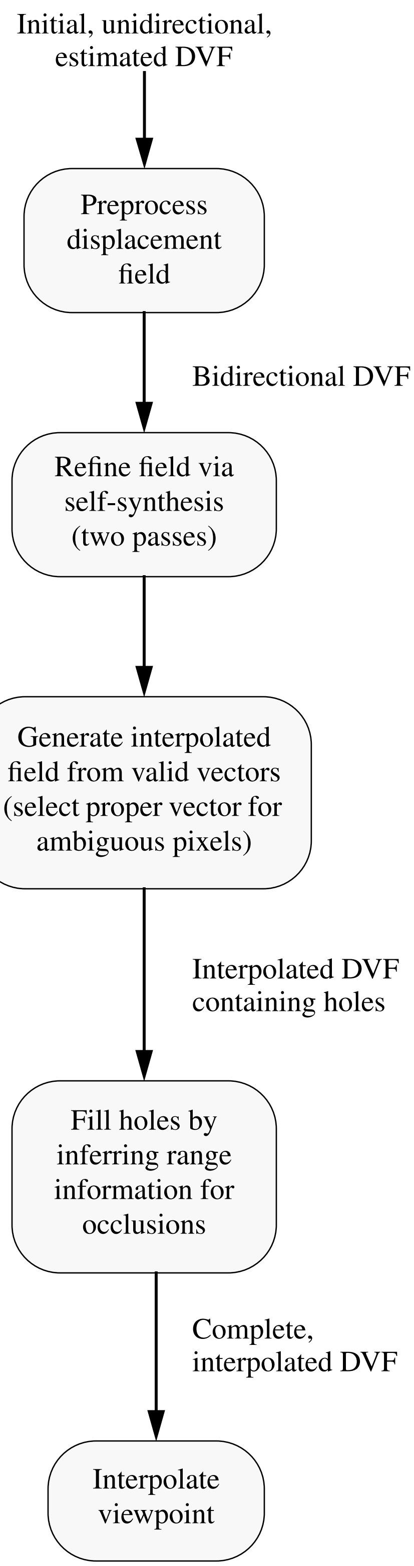


Figure 6.4: Flow-chart for improved viewpoint interpolation technique.

This constraint restricts the epipolar-lines to be parallel with the u-axis of the image frames. We will also assume that Camera 0 was located to the left of Camera 1, i.e., $\Delta c_x > 0$, which provides the required relationship between relative displacement values and the range ordering of objects in the scene.

The v^{th}-row in the preprocessed field, $\hat{\boldsymbol{d}}_{0 \leftarrow 1}(u, v)$, is traversed, starting at pixel location $u = 0$, until either a displacement discontinuity or a pixel with that does not have a valid displacement estimate is found. First, we consider the case of a displacement discontinuity.

We denote the pixel location of the discontinuity by u_L. We define $d_i \equiv \hat{d}_u(u_L - 1, v)$ and $d_f \equiv \hat{d}_u(u_L, v)$, which are the displacement values for the pixels on either side of the discontinuity. Obviously, $d_i \neq d_f$. We assume that these values are the actual displacements for nearby image regions along the epipolar-line, but that the boundary between these unoccluded regions may be incorrect due to the block-based displacement estimation procedure. Therefore, our task is to estimate the true location of the displacement discontinuity.

The field's epipolar-line is traversed from the discontinuity, in the direction of the negative u-axis, to find the initial pixel location, u_i, of the region $[u_i, \ldots, u_L - 1]$ that has constant displacement equal to d_i. The same operation is performed along the positive u-axis to locate the region $[u_L, \ldots, u_f]$, where each displacement estimate equals d_j.[7] Mathematically, the boundaries of these regions are given by,

$$u_i = \arg \min_{u_j} \{\hat{d}_u(u, v) = d_i, \forall u \in [u_j, \ldots, u_L - 1]\} \tag{6.9}$$

$$u_f = \arg \max_{u_j} \{\hat{d}_u(u, v) = d_f, \forall u \in [u_L, \ldots, u_j]\} \tag{6.10}$$

7. A more efficient implementation than repeated traversals of the epipolar-line can be obtained through the use a doubly-linked list of displacement discontinuities. However, we present the self-synthesis algorithm in this manner to simplify the discussion.

We estimate the true location of the displacement discontinuity within $[u_i, \ldots, u_f]$ by calculating the optimal unoccluded region boundary, where the optimization is with respect to generating the minimum-distortion interpolation of View 0 from View 1 for this region.

To reduce the effects of image noise and to exploit the correlation between displacements of neighboring epipolar-lines, we use a distortion function with a region of support that encompasses adjacent image rows. For simplicity, we use the MAD distortion function for the calculation of the optimal displacement discontinuity location. The synthesis distortions for each pixel in the region of interest, obtained from using either displacement values d_i or d_f, are,

$$e_i(u) = \sum_{k \in R_k} h(k) \left| \bar{I}(u, v-k, 0) - \bar{I}(u-d_i, v-k, 1) \right| \tag{6.11}$$

$$e_f(u) = \sum_{k \in R_k} h(k) \left| \bar{I}(u, v-k, 0) - \bar{I}(u-d_f, v-k, 1) \right| \tag{6.12}$$

where $h(k)$ filters the absolute residual between the reconstructed and interpolated pixels along the v-axis over the region of support specified by R_k. The complete synthesis distortion, $D_{ss}(L)$, over the entire region for the discontinuity location, L, is given by,

$$D_{ss}(L) = \sum_{u = u_i}^{L-1} e_i(u) + \sum_{u = L}^{u_f} e_f(u) \tag{6.13}$$

The estimate of the actual discontinuity location is defined as the boundary that minimizes $D_{ss}(L)$:

$$\hat{L}_{opt} = \arg \min_{L \in R_L} \{D_{ss}(L)\} \tag{6.14}$$

where $R_L \in [u_i, \ldots, u_f + 1]$.

We can reduce the complexity of the synthesis distortion calculation from that presented in Eq. (6.13) by realizing that only a single term varies in each of the two summations for successive values of L. In our implementation, we first calculate the synthesis distortion obtained from using the displacement value d_f for the entire set of pixels bounded by u_i and u_f, i.e., $D_{ss}(u_i)$ in Eq. (6.13). The distortions for all other values of L then are obtained from the following difference equation:

$$D_{ss}(L) = D_{ss}(L-1) + e_i(L-1) - e_f(L-1) \tag{6.15}$$

Once the minimum-distortion discontinuity location has been obtained, the displacement estimates within the region of interest are adjusted to reflect the new value:

$$\hat{d}(u, v) = \begin{cases} d_i, & \text{if } u_i \le u < \hat{L}_{\text{opt}} \\ d_f, & \text{if } \hat{L}_{\text{opt}} \le u \le u_f \end{cases} \tag{6.16}$$

A process similar to the adjustment of the discontinuity location is performed when improper holes are detected in the epipolar-line. We denote the pixel location of the first pixel, in a set of contiguous pixels, that do not have valid displacement vectors by u_i. We then traverse the image row in the direction of the positive u-axis until a pixel with a valid displacement estimate is reached. We retain the location, u_f, of the last pixel with an invalid estimate:

$$u_f = \arg\max_{u_j} \{\hat{d}_u(u, v) = \varnothing, \forall u \in [u_i, \ldots, u_j]\} \tag{6.17}$$

where we recall that $\varnothing$ represents an invalid displacement vector. The valid displacement estimates that bound the hole are examined to ascertain whether the hole is improper. The hole must be checked to ensure that it is improper to avoid the assignment of meaningless displacement estimates to actual occlusions.

We define $d_i \equiv \hat{d}_u(u_i - 1, v)$ and $d_f \equiv \hat{d}_u(u_f + 1, v)$.[8] If $d_i \ge d_f$ and there exists a corresponding hole in the reversed field from $[u_i - d_i, \ldots, u_f - d_f]$, we assume that the hole is improper; i.e., it does not correspond to an actual occlusion in the given view. This classification is based on the violation of the principle that states that a discontinuity in the displacement field of

8. If the beginning or ending of the hole occurs at an image boundary, the corresponding displacement value (d_{i} or d_f) is set to zero.

one view corresponds to an occlusion in the other view, whose length, along an epipolar-line, is equal to the magnitude of the discontinuity (see Section 3.2.2).

The improper hole in the displacement field is filled from Eq. (6.16) using an arbitrary boundary between unoccluded regions. If $d_i \neq d_f$, we use the previously described discontinuity adjustment procedure to obtain the optimal location of the new unoccluded region boundary formed by filling the hole.

This process of filling improper holes and adjusting the location of displacement discontinuities is repeated until the entire image row has been traversed. The self-synthesis of the forward field is complete when each row has been processed. The same procedure then is performed on the reversed field.

The process of self-synthesis for a single image row is depicted in Fig. 6.5. The actual displacement field and the estimated field after preprocessing are shown in Figs. 6.5a and 6.5b, respectively. The various steps of the first pass that synthesizes View 0 from View 1 are shown in Figs. 6.5(b)-(d), and the second pass is illustrated in Figs. 6.5(e)-(h). The adjusted and filled displacement values are represented by the dashed lines after each step. We observe that the final, processed field is identical to the actual field. While perfect reconstruction is unlikely in practice, this scenario illustrates the capability of the self-synthesis technique to refine an estimated field, provided that the initial field contains a sufficient number of accurate estimates.

6.3.2 Displacement Selection for Ambiguous Regions

Refining the displacement field through the self-synthesis process alleviates many of the displacement errors that otherwise would have resulted in ambiguously-referenced regions in the interpolated field. Nevertheless, inaccurate displacement estimates may still exist in the processed field. We incorporate both range ordering and a measure of the reliability of an estimate to select the proper displacement for pixels that are referenced by more than one vector.

We denote two displacement vectors that intersect at a pixel location in the interpolated field, $\hat{\boldsymbol{d}}_{\beta \leftarrow 0}(u, v)$, by d_1 and d_2. We use the prediction block-SNR of each of these vectors, SNR_1 and SNR_2, to quantify the reliability of the estimates. The SNR of a displacement estimate is calculated from the variance and prediction distortion of an image block in the decoded reference

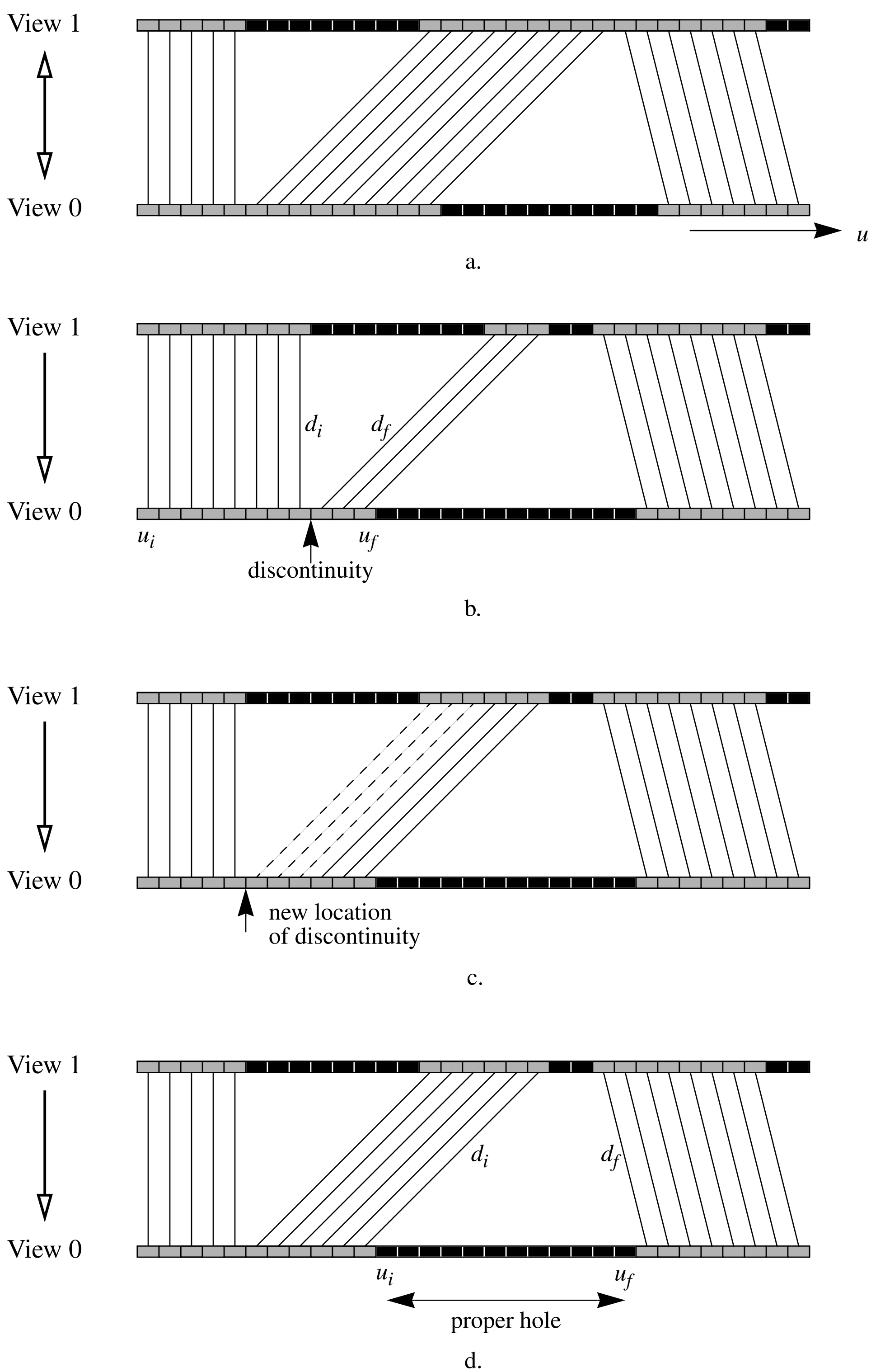
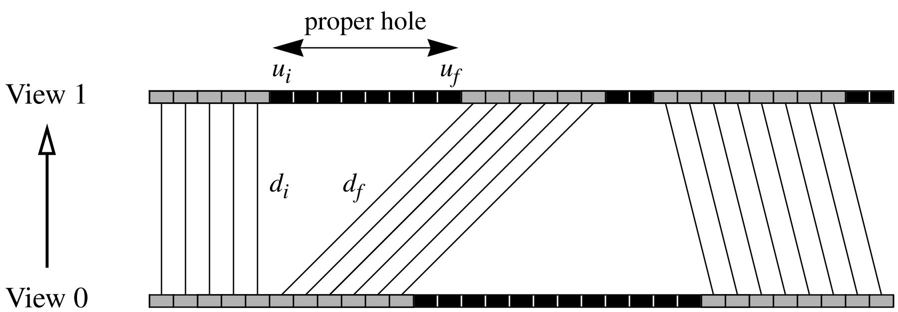
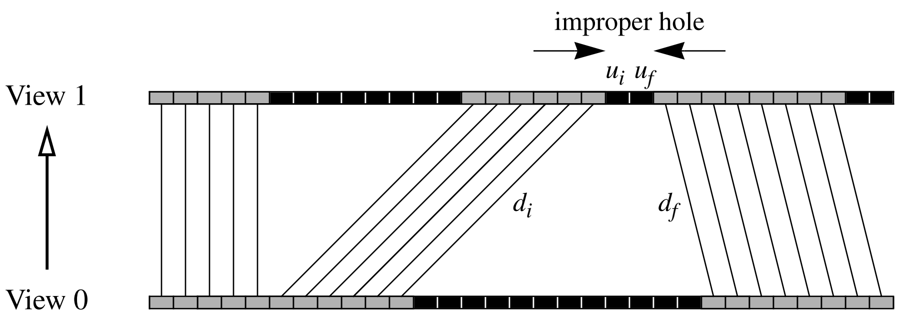


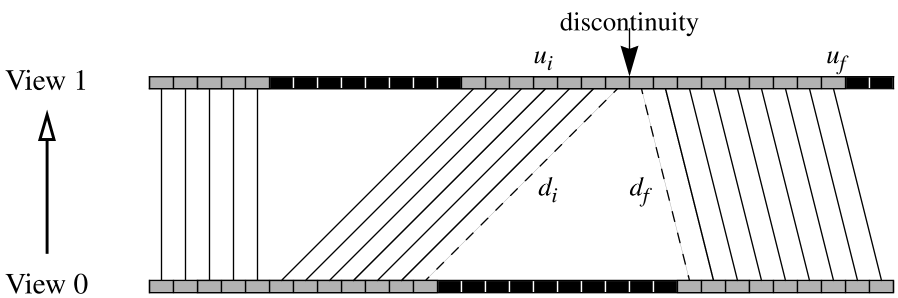
Figure 6.5: Self-synthesis operation along single epipolar-line, (a) actual displacement field, (b)-(d) synthesis of View 0 from View 1, (e)-(g) synthesis of View 1 from View 0, (h) final processed field.



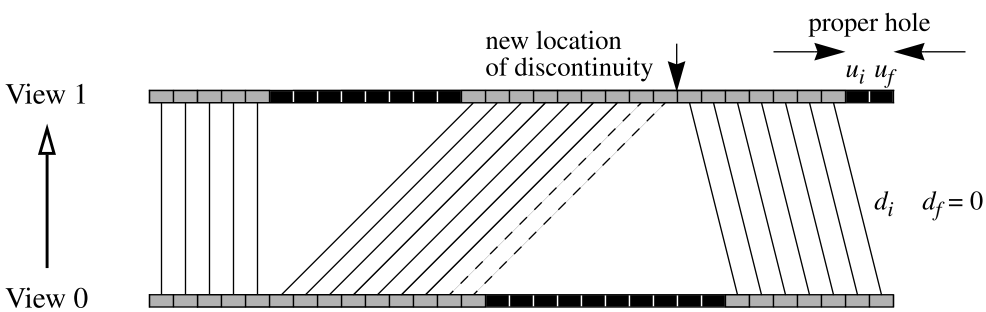
e.



f.



g.



h.

Figure 6.5: (cont.)

frames. We include an additive term, q, to account for quantization noise introduced by the lossy encoder:

$$\mathrm{SNR} = \frac{\hat{\sigma}^2 + q}{D + q} \tag{6.18}$$

where $\hat{\sigma}^2$ is the variance of the reconstructed image region (Eq. (2.11)), and D is the distortion between the predicted and reconstructed block after residual decoding (Eq. (2.9)). The quantization term is obtained from the characteristics of the residual encoder.

The selection of the displacement vector for the ambiguously-referenced pixel is given by,

$$d_{\mathrm{ambig}}(u, v) = \begin{cases} d_1, & \text{if } \left(\dfrac{\mathrm{SNR}_1}{\mathrm{SNR}_2} > T_{\mathrm{high}} \text{ or } \left(\dfrac{\mathrm{SNR}_1}{\mathrm{SNR}_2} > T_{\mathrm{low}} \text{ and } d_1 > d_2\right)\right) \\ d_2, & \text{otherwise} \end{cases} \tag{6.19}$$

where T_{low} and T_{high} are user-specified thresholds that control the contribution of range ordering and estimate reliability on the selection. If $T_{\mathrm{low}} = 0$ and $T_{\mathrm{high}} \to \infty$, this scheme is equivalent to the common approach of using only range ordering. If more than two displacement estimates intersect at a single pixel in the interpolated field, a binary search is performed for the likely correct estimate using Eq. (6.19).

6.3.3 Inferring Range Information for Occlusions

All holes remaining in the interpolated field are assumed to proper holes, i.e., they are due to occlusions in the reference views. We assign displacement vectors to these regions based on inferred range information for the occlusion. We again consider the case of two, horizontally-offset cameras, where C_0 is positioned to the left of C_1.

We assume that the scene has a *regular* structure, in that occluded regions have constant depth, and that this depth is equal to the depth of the background unoccluded region adjacent to the occlusion. While one can imagine situations were these assumptions are violated, our experience is that the majority of real-world scenes have regular structures.

The interpolated field, $\hat{\boldsymbol{d}}_{\beta \leftarrow 0}(u, v)$, is traversed along each row. When a pixel is reached that does not have a valid displacement estimate is reached, the initial location of the hole is noted (u_i).

The final pixel location of this hole then is obtained from Eq. (6.17), which we have re-written for convenience:

$$u_f = \arg \max_{u_j} \{\hat{d}_u(u, v) = \varnothing, \forall u \in [u_i, \ldots, u_j]\} \tag{6.20}$$

The displacements values on either side of the hole, $d_i \equiv \hat{d}_u(u_i - 1, v)$ and $d_f \equiv \hat{d}_u(u_f + 1, v)$, are examined to calculate the proper mapping for the hole and to resolve its visibility.

For the given camera geometry, the range of a scene object is related directly to its displacement – the larger the displacement, the farther the object is from the camera pair. Therefore, the background unoccluded region adjacent to the hole is the region with maximum displacement. We assign pixels in the hole using,

$$d_{\text{hole}}(u, v) = \begin{cases} d_i, & \text{if} \quad d_i > d_f \\ d_f, & \text{otherwise} \end{cases} \tag{6.21}$$

Also, if $d_i > d_j$, the geometry of the scene indicates that the occlusion was visible in View 0; otherwise, it was visible in View 1. We mark each pixel in the hole as to in which view the region was visible. This information is used in the final interpolation of the pixel intensity from Eq. (6.7).

A special case for this procedure occurs when one of the end-points of the hole is located at an image border. If $u_i = 0$, we assign the displacement d_f to the pixels in the hole. The visibility of the occlusion is given by: visible in View 0 if $d_f < 0$, or visible in View 1 if $d_f > 0$. Appropriate modifications are performed when the hole occurs at the right image border, i.e., $u_f = U - 1$.

Figure 6.6a depicts a simple scene consisting of three planar objects, in one dimension. The perspective projection of an object for each view is given by the intersection the projection plane with the projection from the object through the center of projection (camera lens). Interesting projections are drawn in the figure, where the objects have been shaded according to whether a particular portion is visible to one or both of the cameras. The tic-marks along the projection plane represent pixels in the image frames.

The displacement field relating corresponding points in Views 0 and 1 for this sample scene is shown in Fig. 6.6b. The interpolated field obtained by mapping unoccluded regions also is pro-

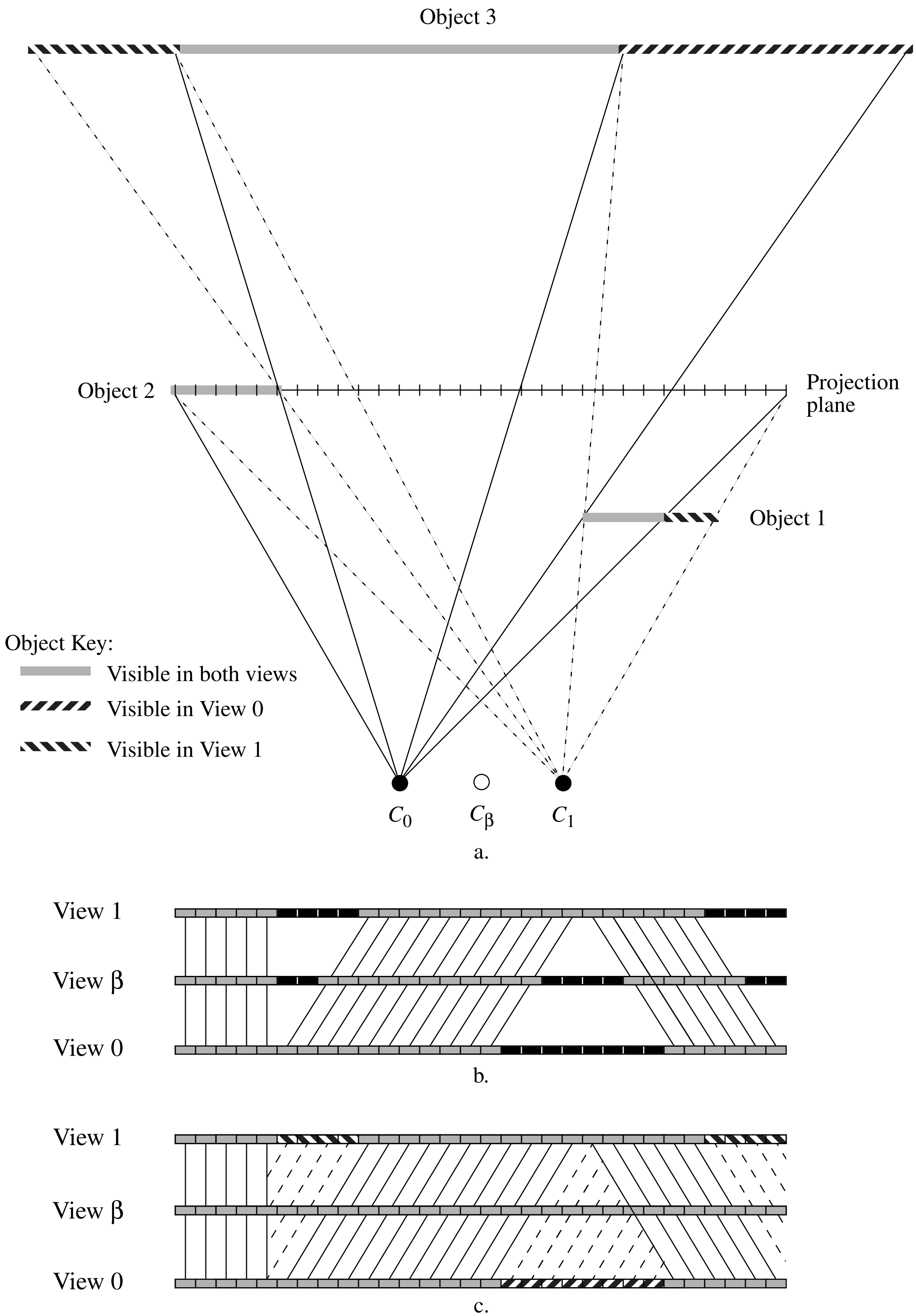


Figure 6.6: Inferring range information for occluded regions, (a) perspective project of sample scene, (b) field interpolated from unoccluded regions, (c) interpolated field after filling occluded regions.

vided, where the holes in the interpolated field are marked by the blackened pixels. Figure 6.6c illustrates the results of the described algorithm – the dashed lines represent the inferred displacement vectors for occluded regions. Since this scene has a regular structure, the inferred displacement vectors corresponds exactly to the true object displacement.

6.4 Cost Analysis

In this section, we quantify the complexity and storage costs of our improved viewpoint interpolation technique. When applicable, we make comparisons to the interpolation schemes described in Section 6.2.

Before we provide our cost analysis, we make two observations on interpolation techniques. The first observation is that the complete synthesis operation can be broken down into the distinct steps of generating a *dense* interpolated displacement vector field[9], and interpolating pixel intensities using this field. All of the interpolation schemes known to the author perform the latter step in roughly the same manner; therefore, we will not consider this process in our complexity comparisons. Secondly, many of the previously reported algorithms have not been developed in the context of a complete multi-view encoder/decoder system; they do not indicate if the dense field is generated at the receiver or transmitter, and they do not consider the cost of transmitting the field if it was estimated at the transmitter. To provide the most meaningful comparison, we will make reasonable assumptions for the techniques that fail to mention these issues.

6.4.1 Complexity

Our technique generates the dense interpolated field by processing an initial displacement field, obtained from a block-based procedure. Since we assume that the initial field is extracted from the coded bit-stream, its calculation is essentially free, with respect to the interpolation operation. For completeness, we denote the encoder complexity of our new algorithm by,

$$C_{\text{new-encoder}} = C_{\text{BMA}} \tag{6.22}$$

where C_{BMA} is the complexity of the block-matching algorithm, which is on the order of the displacement estimation search range for a full-search technique.

9. The field is considered dense if it contains a displacement vector for each pixel location in the image.

Procedure	Operations-per-pixel
Preprocessing	5
Unoccluded regions	4
Ambiguous regions	< 1
Occluded regions	< 1

Table 6.1: Average fixed complexity costs for the generation of interpolated displacement vector field.

The initial field is processed at the decoder to: 1) eliminate likely false estimates (preprocessing), 2) adjust the location of displacement discontinuities and fill improper holes (self-synthesis), 3) select the displacement for ambiguously-referenced pixels, and 4) fill proper holes by inferring range information for occlusions. Except for the self-synthesis procedure, each of these steps requires a fixed number of operations-per-pixel (see Table 6.1). The total decoder complexity of the our method is:

$$C_{\text{new-decoder}} = 2C_{\text{ss}} + 11 \tag{6.23}$$

where C_{ss} represents the number of operations for the unidirectional self-synthesis procedure, and the constant term equals the maximum number of operations for the steps with fixed complexity.

The complexity of the self-synthesis operation is difficult to analyze since it depends on the number of discontinuities and improper holes in the initial field. Using Eq. (6.16) reduces the complexity of locating the optimal unoccluded region boundary to a linear search over $[u_i, \ldots, u_f + 1]$. Experimental results indicate that both passes of the self-synthesis procedure require an order of magnitude fewer operations than the full-search, block-based displacement estimation step. This is particularly impressive since the search range for spatially-offset views is one-dimensional. Our technique, therefore, is applicable to systems with a low-complexity decoder constraint.

The EPI-based techniques generate the dense interpolated field using numerous, closely-spaced views of the scene; the individual techniques used vary from nine [47] to 180 views [40]. Hence, we assume that the displacement field relating corresponding points within the extreme views is generated at the encoder. The interpolated field is calculated at the decoder based on the relative location of the desired viewpoint. The number of operations required to handle occlusions and ambiguously-referenced regions for each of these techniques are approximately equal to the last three rows in Table 6.1, i.e., $C_{\text{EPI-decoder}} = 6$.

Hsu, *et. al.*, use an edge detection and region growing procedure to estimate the displacement field. Although the exact method was not described, region growing is typically considered a difficult (read computationally costly) procedure [40].

In [47], the interpolated field is generated by searching for the minimum variance trace line for each pixel. To constrain the search, minimum and maximum trace line slopes are specified. For an EPI containing *M* views, the per-pixel encoder complexity of this method is given by,

$$C_{\text{Katayama-encoder}} = 2M(d_{\max} - d_{\min} + 1) \tag{6.24}$$

where $d_{\min}$ and $d_{\max}$ at the relative locations of the end-points for the trace lines with the extremum slope values.

The final EPI-based method of Fujii and Harashima requires an enormous search space for the calculation of the triangular mesh grid points that yield the minimum *Var*-space. The authors did not provide the search boundaries, so we cannot directly calculate the number of operations-per-pixel; however, they did indicate the processing time required to perform their algorithm on a Silicon Graphics Iris Indigo Elan.[10] Without the *Var*-space segmentation enhancement, fields for CIF-sized images were obtained in “a few minutes”; with segmentation, the computational time was “about 10 minutes” [30]. Obviously, this approach has excessive complexity.

The three-camera technique developed by Skerjanc and Liu generates the interpolated field through: feature extraction, feature correspondence matching, and correspondence compatibility. Due to the object-oriented (i.e., feature-based) approach, it is difficult to compute per-pixel complexity quantities. Since each of these steps are performed in triplicate – once for each view or pairs of views – we feel that this method is considerably more complex than our technique.

The technique presented by Thoma and Bierling is the most similar to our method in terms of the partitioning of operations and complexity. An initial displacement field is obtained at the encoder using a hierarchical block-matching algorithm (H-BMA):

$$C_{\text{Thoma-encoder}} = C_{\text{H-BMA}} \tag{6.25}$$

10. Unfortunately, no mention of the processor or clock speed was given.

The field then is processed at the decoder to classify covered and uncovered regions using: change detection, median filtering of a binary classification mask, and elimination of small regions. The change detection and median filtering operations are sufficiently well described that we can calculate the complexity of these steps; however, the explanation on how small regions are eliminated was left somewhat open-ended. For the values provided in Table 3 of their work, the decoder complexity is:

$$C_{\text{Thoma-decoder}} = C_{\text{ESR}} + 26 \tag{6.26}$$

where C_{ESR} is the number of operations-per-pixel to eliminate small regions and the constant term is due to the change detection (9 operations), median filtering (13 operations), and mapping of unoccluded pixels (4 operations).

Cafforio's, *et. al.*, method for motion-compensated interpolation performs a pel-recursive calculation of the forward and backward displacement fields between temporally-offset frames. To improve the robustness of the field, the recursion is performed four times for each field:

$$C_{\text{Cafforio-encoder}} = 8C_{\text{pel-recur}} \tag{6.27}$$

where $C_{\text{pel-recur}}$ represents the number of operations-per-pixel for the pel-recursive analysis. Due to the iterative process to generate the initial fields, we cannot assign a definitive quantity to this variable. At the decoder, the bidirectional fields are processed to detect occluded regions and fill pixels un-interpolated pels. Since the authors claim that only 2% of the pixels in the interpolated field do not have valid displacement estimates, we will not include the filling of these pixels in the complexity analysis. Finally, a 5-point median filter is applied to the interpolated field to ensure spatial congruence in low detail areas. Assuming that the occlusion detection uses a window size of 3×3 pixels, the total complexity of the decoder is then,

$$C_{\text{Cafforio-decoder}} = 19 \tag{6.28}$$

which is broken down into: 9 operations for the occlusion detection, 4 operations for the mapping of valid vectors, and 6 operations for the median filter.

The interpolation algorithm by Ribas-Corbera and Sklansky is very similar to that of Cafforio, *et. al*. Here, bidirectional fields are generated at the encoder using optical flow analysis:

$$C_{\text{Ribas-Corbera-encoder}} = 2C_{\text{flow}} \tag{6.29}$$

The fields then are analyzed at the decoder to detect occluded regions (using Cafforio's searching algorithm), and suppress gaps (holes) and ambiguities. The filling of gaps requires two comparisons, while ambiguities are handled through range ordering, i.e., a single comparison of the intersecting displacements. The total decoder complexity, including the mapping of unoccluded displacements, is given by,[11]

$$C_{\text{Ribas-Corbera-decoder}} = 16 \tag{6.30}$$

In summary, our technique is substantially less complex than the EPI-based algorithms. While our decoder requires slightly more operations than that of [13] or [83], our block-based displacement field is encoded considerably more efficiently than their multiple, pixel-based fields; hence, our approach is more applicable to viewpoint interpolation within a complete encoder/decoder system.

6.4.2 Storage

At the encoder, our technique must store the two reference views and a *sparse* displacement field. The field is sparse since the displacement of each $K \times L$ image block is described by a single vector.

At the decoder, additional frame memories are needed to store all of the desired intermediate views. For the binocular motion parallax application, this amounts to a total of four frame stores – two reference views and two interpolated views. We also require two dense displacement fields. The first field relates corresponding pixels between Views 0 and 1, which is obtained by replicating the block vectors in the sparse field to each pixel, followed by the preprocessing and self-synthesis operations. The second displacement field is the actual interpolated field used to predict the

---

11. The authors of this technique felt that the procedure presented by Thoma and Bierling was of "much higher complexity than [their] algorithm and Cafforio's" [83]. In light of this discussion, we believe this characterization is unjustified, unless the "elimination of small regions" in [93] constitutes a significant computational burden.

pixel intensities in the desired view. The data set for each pixel in these fields includes: a binary flag indicating whether the pixel has a valid vector, along with the displacement vector and block-SNR, if valid.

Our storage requirement is approximately equal to that of the various viewpoint interpolation schemes that use only two views. Only our method requires SNR values for each valid vector. Appropriate increases to the total storage cost are experienced if a technique uses bidirectional fields [13, 83] or a change detection operator [93]. The other techniques that process more than two views at the encoder require proportionally more storage based on the total number of views [30, 40, 47, 89].

6.5 Experimental Results

The improved interpolation technique was used to synthesize intermediate viewpoints between numerous views in the compressed multi-view sequences described in Section 2.4. The initial displacement vector fields were extracted from the coded bit-stream that described the predictive coding of one of the reference views from the other. The reference views were predicted using a full-search, block-based technique with full-pixel accuracy, where the displacement vectors were estimated using an open-loop structure [68]. Block sizes of either 16×16 or 16×8 pixels were used, depending on whether the view was a complete image frame or only a single field, i.e., sampled at twice the frame rate. The reconstructed quality of the reference views was such that they were indistinguishable from the original views.

For the self-synthesis operations, we used filter taps of $h(-1) = h(1) = 1$ and $h(0) = 2$ in Eqs. (6.11) and (6.12). Also, the thresholds for the selection of ambiguously-referenced regions were set to $T_{\text{low}} = 0.95$ and $T_{\text{low}} = 1.05$. Nominally, all of the reference views were offset in only their horizontal positions. Since exact camera alignment could not be guaranteed, we allowed for some vertical displacement by slightly modifying our procedure to include deviations from the epipolar-line constraint. The self-synthesis operation was still performed along scan-lines, but we perturbed the locations of vertical displacement discontinuities to avoid long stretches of horizontal holes.

The first results were obtained for the *Flower Garden* sequence. Although this is actually a single-view sequence, the scene objects are stationary and the camera motion is almost completely horizontal – two temporally-offset frames satisfy our camera geometry constraint. We illustrate the interpolation results from using frames 0 and 3 as the reference views (Fig. 6.7). The major occlusions in frame 0 are to the left of the large foreground tree and the left image border, and vice versa for frame 3. The occluded portions are estimated to occupy roughly 7% of the image frames. The true horizontal displacement in this pair of images was calculated to vary from 0 to 20 pixels.

Figure 6.8 illustrates the horizontal component of the displacement fields relating frames 0 and 3 after the various processing steps of our algorithm. We observe the poorly delineated object boundaries and improper holes in the preprocessed field. After self-synthesis of frame 0 from frame 3, Fig. 6.8b, the displacement discontinuity along the right side of the tree closely matches the true object boundary. Similar refinement for the left side of the tree is obtained from the second self-synthesis pass; the blockiness of the initial field is completely eliminated.

The blackened pixels in the displacement fields represent pixels assumed to be occluded. The corresponding pixels in the original frames have been blackened in Fig. 6.9 to examine the performance of this method to detect occlusion. The major occlusions are accurately localized. Most of the spurious invalidated pixels are due to displacement errors in regions with uniform intensity, e.g., the sky, and should not detract from the subjective quality of the interpolated views.

Frame 1 was interpolated using the processed field by setting $\beta = \frac{1}{3}$. The interpolated view is virtually indistinguishable from the original (Fig. 6.10). Accurate synthesis of the image borders and sides of the foreground tree were obtained, and a PSNR of over 28 dB was measured.[12] This represents an improvement of over 3.5 dB from our earlier work [63], which did not include the preprocessing and self-synthesis steps. Fujii and Harashima reported only 17 dB for the interpolation of a similar frame in this sequence [30].[13] The absolute frame difference from using the repitition of frame 0 and our interpolation technique for this image are shown in Fig. 6.11. The frame repitition results indicate that a majority of the image is displaced between frames.

For comparison, we interpolated this frame using the unprocessed, block-based displacement field, where the displacement of ambiguously-referenced pixels were chosen based solely on range ordering (Fig. 6.12a). Since the common assumption of zero displacement for holes in the interpolated field is violated for spatially-offset views, we filled these holes by using our method of inferring range information for occluded regions. We observe very poor reconstruction of the image borders and the regions surrounding the foreground tree. We also include the results of performing

12. We note that even better performance could have been obtained if we were provided with the exact camera motion between frames, as opposed to the assumed motion – calculation of the exact epipolar-line would be possible and the relative location of the intermediate frames would be known.

13. The authors did not distinguish between SNR and PSNR in their work. Based on the subjective quality of the reconstructed image, we feel that the 17 dB quantity represented PSNR.

a bidirectional block-based prediction of this frame using frame 0 and 3 as the reference frames in Fig. 6.12b. While the PSNR of this image is comparable to the interpolated view using our improved technique, the subjective quality of the interpolated image is far superior. Also, over 4000 bits are required to describe the bidirectionally-predicted image, while no bits need to be transmitted for the interpolated view.

The results obtained for the interpolation of frame 2, $\beta = \frac{2}{3}$, are illustrated in Figs. 6.13-6.15. The same procedure was performed for the entire Flower Garden sequence, where the reference views were offset by three frames, i.e., 3:1 temporal subsampling. The average PSNR for the interpolated views was approximately 27.5 dB. The interpolation was simulated in C on a Silicon Graphics workstation with a single R4400, 174 MHz processor. Each interpolated displacement vector field was generated in less than 1.5 seconds.

A total of sixty views, obtained from the interpolation of 20 views between each of first three pairs of reference views, have been used extensively in a binocular multi-view display system. A head-tracking device is used to display the appropriate stereo-pair based on the viewer's horizontal position, and a "stereo-dial" allows viewers to specify the inter-camera separation. Although some regions of the interpolated views exhibit inconsistent object displacement due to unrectified errors in the displacement field, viewers report effective motion parallax and diminished eye strain with this system.

The second result that we will examine is the viewpoint interpolation of the two-view *Volleyball* sequence. The original 155$^{\text{th}}$ frames for each view of this sequence are shown in Fig. 6.16. We observe that the reference views are very similar – occluded regions are almost totally non-existent and when an occlusion does exist it has uniform intensity, e.g., the occlusions between the players and the ground. We manually calculated the true range of the horizontal displacement to equal 14 pixels.

Interpolation of the viewpoint located halfway between the reference views was performed using our improved technique and a basic interpolation scheme (Fig. 6.17). The basic interpolator uses the unprocessed, block-based displacement field, selects the displacement for ambiguous regions using range ordering, and assigns zero displacement to holes in the interpolated field using the technique described in [83]. Both methods achieve subjectively-pleasing interpolated views for this sequence. These results indicate that the basic interpolator is sufficient for scenes with relatively simple structures and limited occlusion. The real advantage of our improved technique is evident for more complex scenes, as illustrated for the following sequence.

Numerous intermediate views were synthesized between Views 1 and 2 of the *Cement Mixer* sequence.[14] We will examine the results for the 175th frame, where the original images are illustrated in Fig. 6.18. Large occlusions exist on either side of the foreground tree and image borders, and the scene objects are located at numerous depth planes; less obvious occlusions are present surrounding the person located between the cars and truck. The occluded portions are estimated to occupy roughly 10% of the reference views. The horizontal displacement range was estimated to be over 50 pixels, which is substantially greater than any scene used to evaluate prior interpolation techniques known to the author.

The intermediate viewpoints, $\beta = \left[\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\right]$, obtained from using our technique are shown in Figs 6.19, 6.20, and 6.21, respectively. While visual artifacts are present, we feel that these interpolated views are quite impressive given the complexity of the scene. In particular, the image borders are faithfully reconstructed and appropriate regions behind the foreground tree become visible as β varies; e.g., the entire ladder is visible in $F(1.75,175)$, but only a portion of it can be seen in $F(1.25.175)$.

We compare these results with those obtained by the basic interpolator by eliminating various enhancements from our procedure. The interpolated view shown in Fig. 6.22 was generated using the initial, unprocessed displacement vector field. Ambiguous regions and holes in the interpolated field were handled using our method. The right image border is poorly interpolated, and objectionable block artifacts are present throughout the intermediate view. The next comparison was made by interpolating the same viewpoint using the displacement field after preprocessing and self-synthesis, but without our ambiguous regions selection and occlusion filling procedures (Fig. 6.23). The blockiness of the previous scheme is eliminated, but artifacts resembling speckle noise now exist at object boundaries. Also, many of the regions in the interpolated view do not correspond to the true geometry of the scene; the end of the cement chute is visible on the right side of the tree and the majority of the ladder is missing. These errors are due to the improper assumption of zero displacement for holes in the interpolated field. Figure 6.24 illustrates the interpolation performance of the basic scheme that does not include any of our enhancements. This unsatisfactory intermediate view is plagued by both blockiness and incorrect scene geometry.

The final results obtained for the *Piano* sequence illustrate the subtleties of our enhanced interpolation scheme. The original 25th frame for Views 0 and 1 of this sequence are shown in Fig.

14. The interpolation of viewpoints between Views 0 and 1 of this sequence were not performed since the respective foci were so poorly aligned.

6.25. While there is a moderate amount of object depth discontinuity, the majority of occlusions are regions of uniform intensity, e.g., the background surrounding the person's head. For a more interesting occlusion, we direct the reader to examine the lower-left corner of $F(0{,}25)$, where a portion of the flower arrangement is occluded by the piano player.

The interpolated viewpoints, located at $\beta = \frac{1}{2}$, obtained from using our enhanced scheme and the basic scheme are shown in Fig. 6.26. Both of these techniques provide subjectively-pleasing results for a majority of the image. Upon closer examination, though, we observe the superiority of our technique. Figure 6.27 contains close-ups of the lower-left corner of the reference and interpolated views. The object to the left of the person, which is only partially visible in $F(1{,}25)$, is accurately reconstructed in Fig. 6.27b. While such improvements appear minor when viewing individual frames, they are significant when the interpolated views are used in the motion parallax and viewer-specified camera separation applications to ensure object continuity.

6.6 Summary

In this chapter, we have presented an improved displacement-compensated interpolation scheme. The interpolation problem was considered in the context of a complete multi-view encoder/decoder system, capable of providing the viewer with simulated motion parallax and control over the perceived depth range in the imagery. To ensure compliance with current predictive coding standards, the interpolation was performed using a single, noisy displacement vector field obtained from a conventional block-based displacement estimation technique.

Errors traditionally associated with block-based displacement estimation schemes were reduced through a preprocessing step to eliminate likely false estimates and a self-synthesis procedure to refine displacement discontinuities and fill improper holes. By using the initial field as a guide as to how the field may be adjusted, pixel-based accuracy for unoccluded region boundaries and occlusions were obtained via the self-synthesis operation with minimal computational complexity.

Pixels assumed to be unoccluded in the processed displacement field were mapped to the intermediate image using the geometric relationship between the views. We used a combination of both range ordering and a displacement estimation confidence measure to handle the situation when a pixel in the intermediate view was mapped by more than one reference image pixel. Un-interpolated pixels were assumed to be due to occlusions in the reference views. To fill these pixels, we inferred range information for occlusions based on a regular-scene-structure assumption.

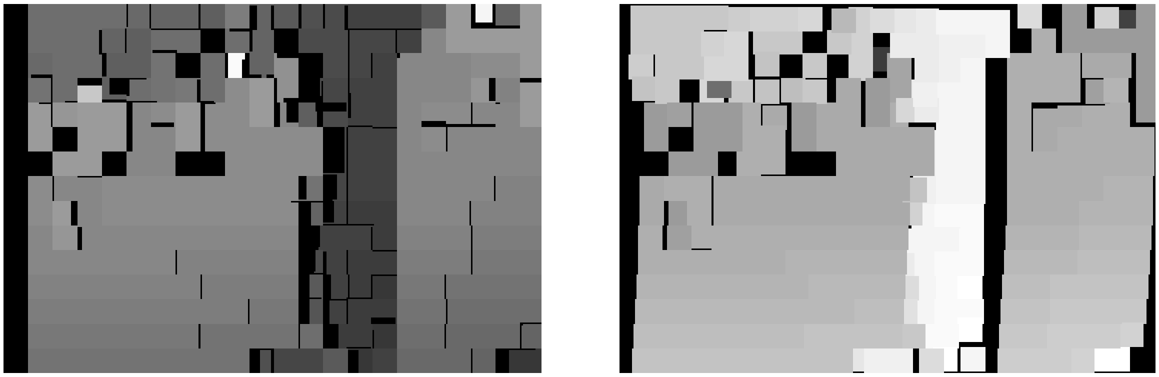
Interpolation results for numerous multi-view signals were presented. Our technique synthesized subjectively-pleasing intermediate views from reference views with moderate to high percentages of occlusions, and required relatively low complexity and storage costs. For the one sequence were the actual intermediate views were available, our improved method achieved an average PSNR of approximately 27.5 dB without requiring any additional bit rate beyond that needed to encode the two reference views. These results validate our claim that satisfactory interpolation is possible from only two spatially-offset views.

Future work includes incorporating additional image information into the self-synthesis procedure to further improve the localization of displacement discontinuities and occlusions. This may entail the integration of edge and color information into the optimal boundary cost function.

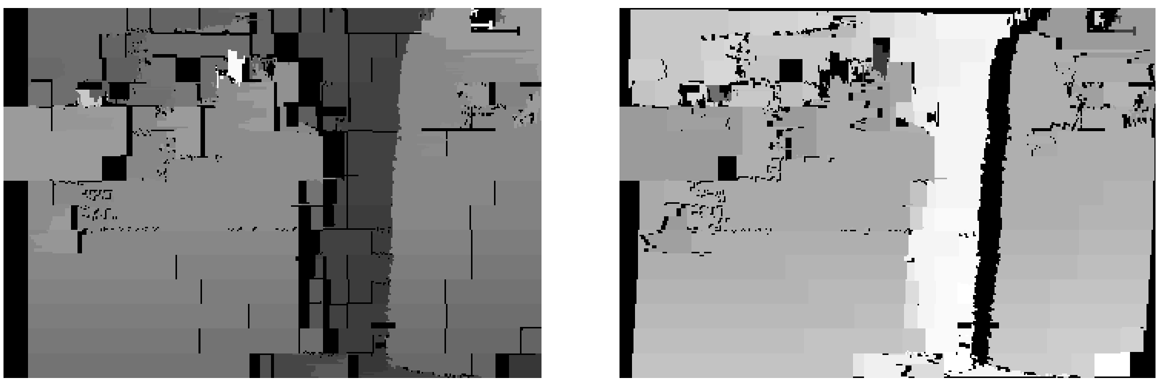
a.

b.

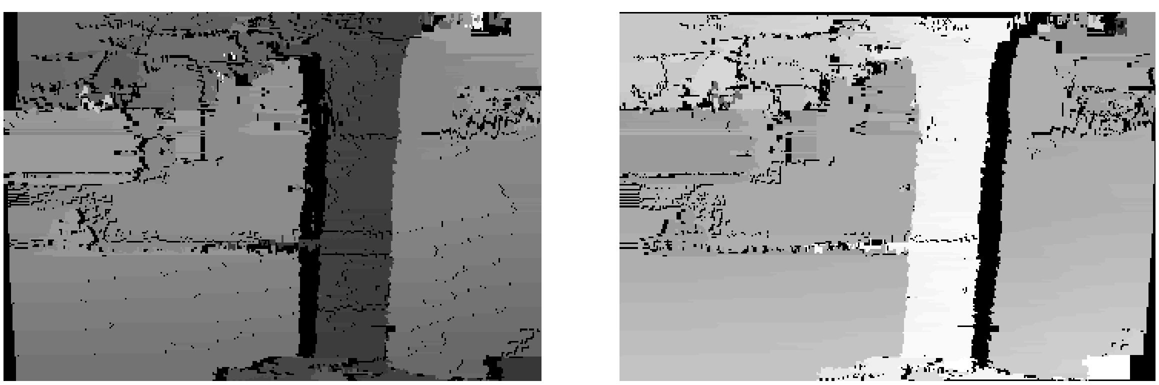
Figure 6.7: *Flower Garden*, (a) original frame 0, (b) original frame 3



a.



b.



c.

Figure 6.8: Estimated displacement fields (horizontal component) after various processing stages. Left column: frame 0 from frame 3. Right column: frame 3 from frame 0. (a) after thresholding, border elimination and field reversal, (b) after self-synthesis of frame 0, (c) after self-synthesis of frame 3.

a.

b.

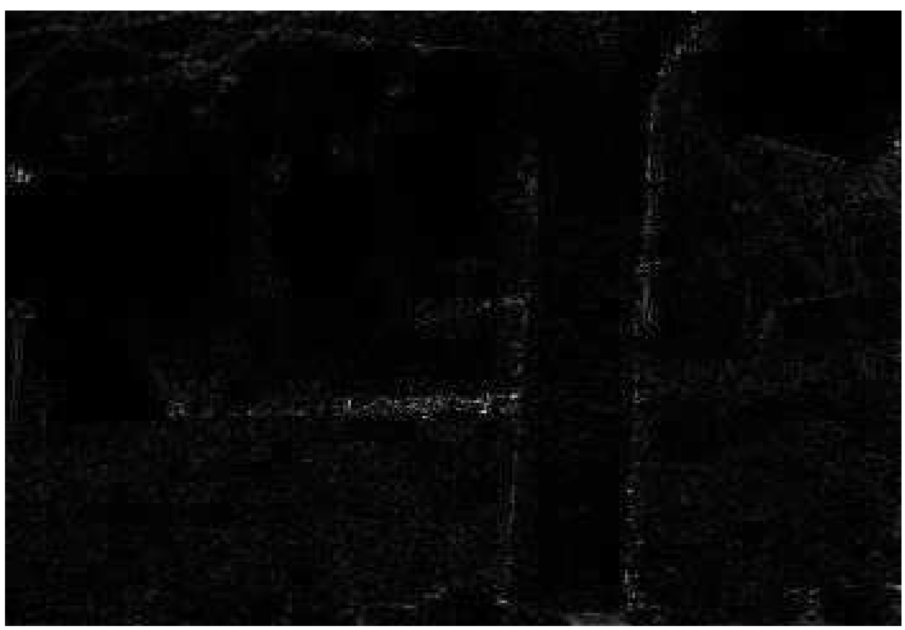
Figure 6.9: Occlusion localization results, (a) frame 0, (b) frame 3.

a.

b.

Figure 6.10: *Flower Garden* frame 1, (a) original, (b) interpolated.

a.



b.

Figure 6.11: Absolute frame difference for frame 1, (a) frame repetition, PSNR=15.72 dB, (b) interpolation, PSNR=28.14 dB.

a.

b.

Figure 6.12: *Flower Garden* frame 1, (a) interpolated without performing displacement field processing and ambiguous region selection (PSNR=24.54 dB), (b) bidirectionally-predicted from reference frames 0 and 3 using a fixed-size, block-based technique (PSNR= 28.55 dB).

a.

b.

Figure 6.13: *Flower Garden* frame 2, (a) original, (b) interpolated.

a.

b.

Figure 6.14: Absolute frame difference for frame 2, (a) frame repetition, PSNR=15.85 dB, (b) interpolation, PSNR=28.00 dB.

a.

b.

Figure 6.15: *Flower Garden* frame 2, (a) interpolated without performing displacement field processing and ambiguous region selection (PSNR=24.86 dB), (b) bidirectionally-predicted from reference frames 0 and 3 using a fixed-size, block-based technique (PSNR= 28.89 dB).

a.

b.

Figure 6.16: *Volleyball*, (a) original $F(0,155)$, (b) original $F(1,155)$

a.

b.

Figure 6.17: *Volleyball F*(0.5,155), (a) interpolated using enhancements, (b) interpolated without enhancements.

a.

b.

Figure 6.18: *Cement Mixer*, (a) original $F(1,175)$, (b) original $F(2,175)$

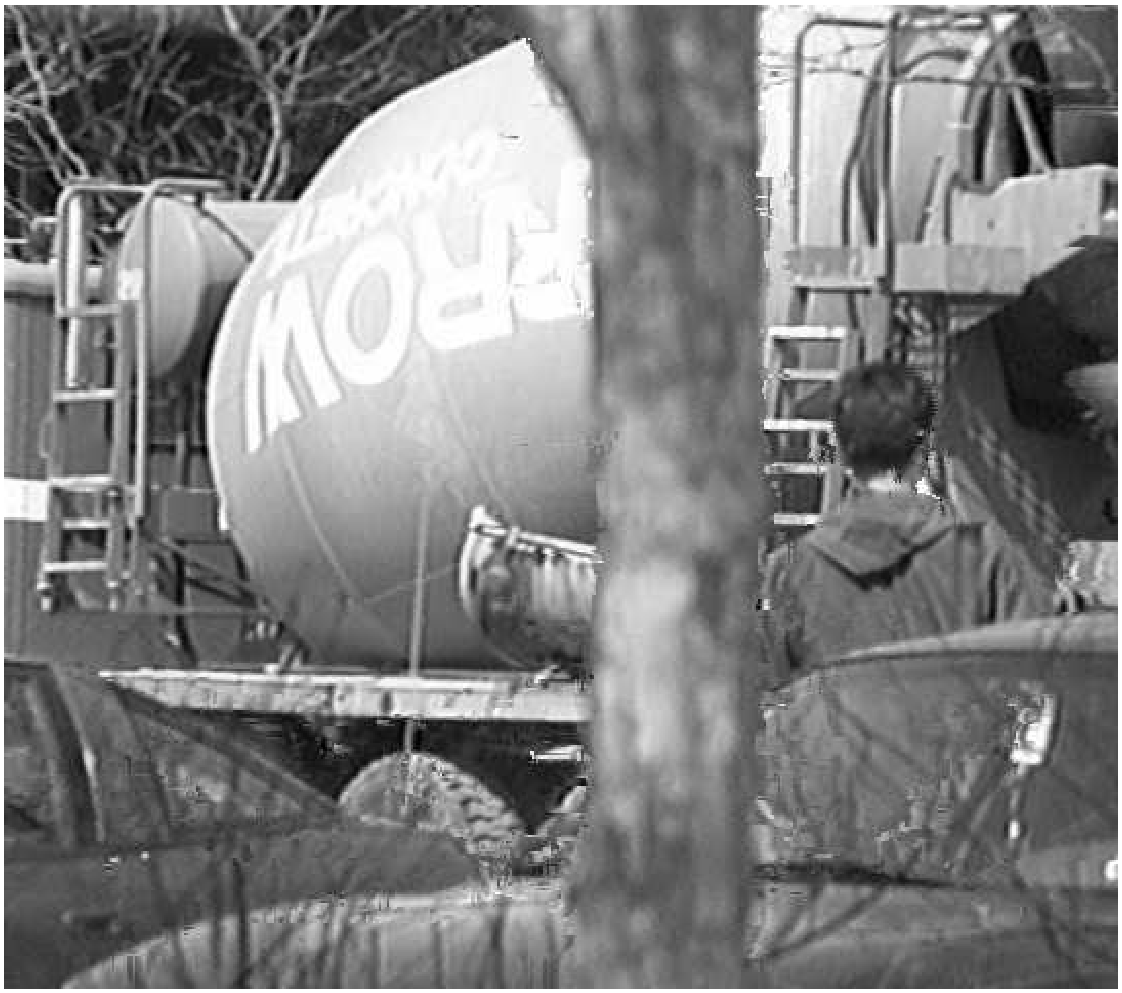


Figure 6.19: *Cement Mixer F*(1.25,175) interpolated using improved technique.

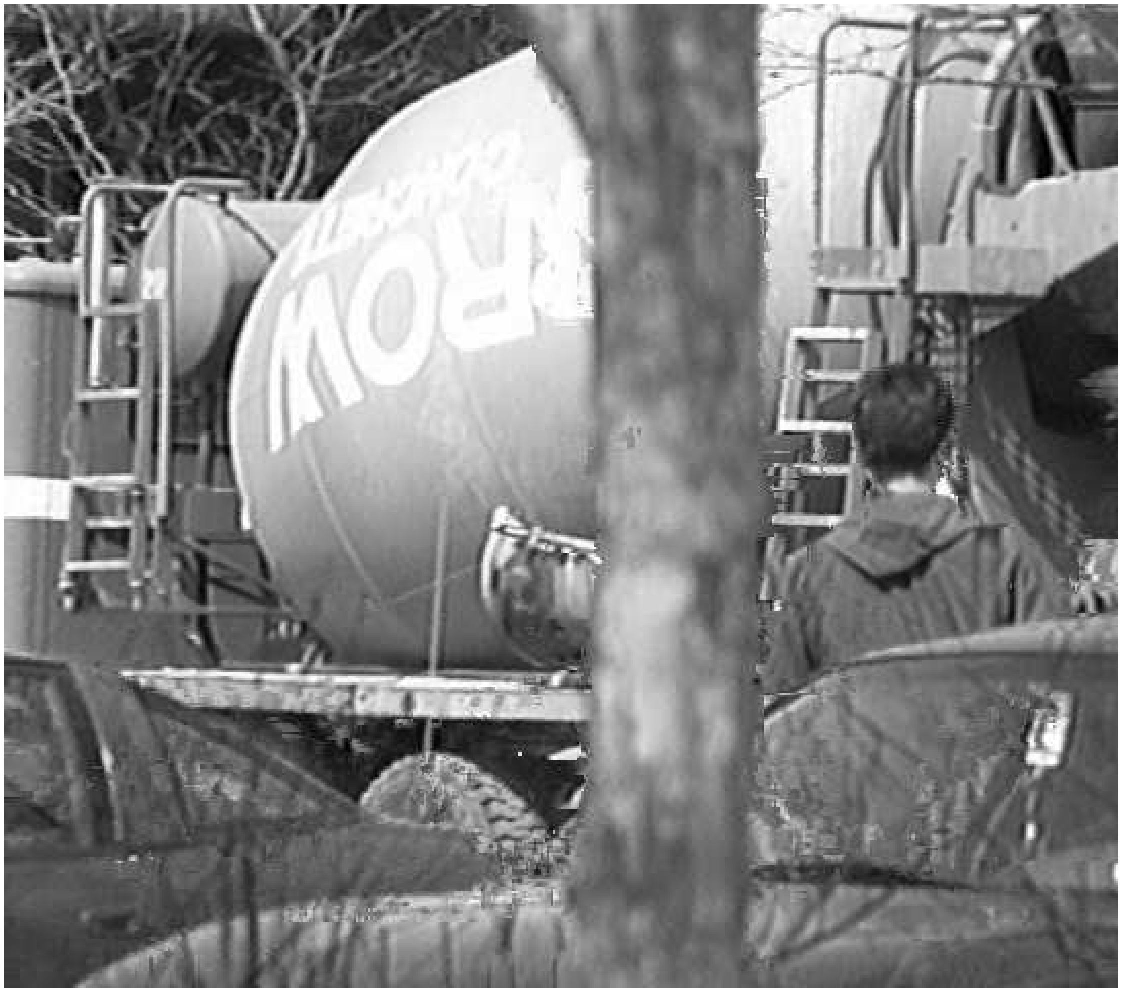


Figure 6.20: *Cement Mixer F*(1.5,175) interpolated using improved technique.

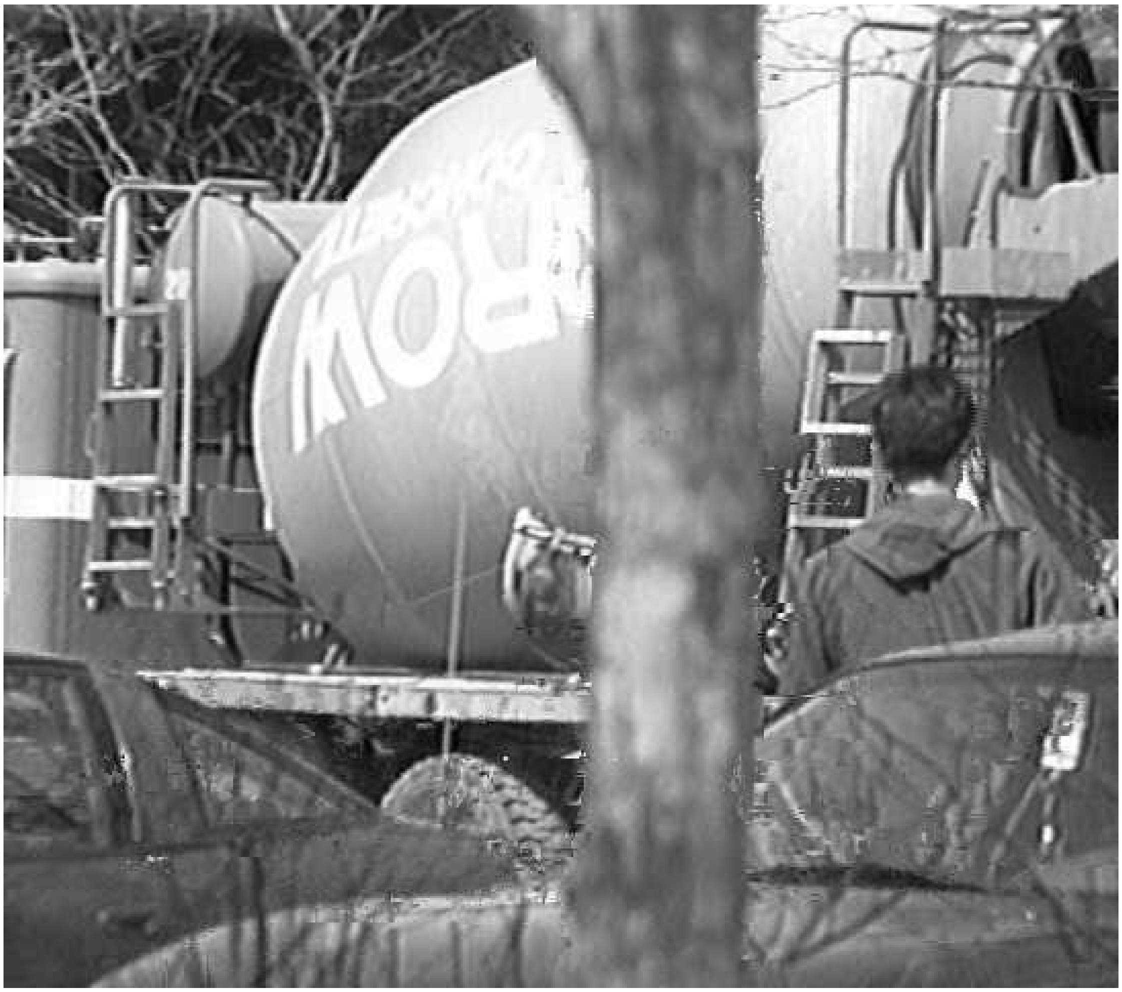


Figure 6.21: *Cement Mixer* *F*(1.75,175) interpolated using improved technique.

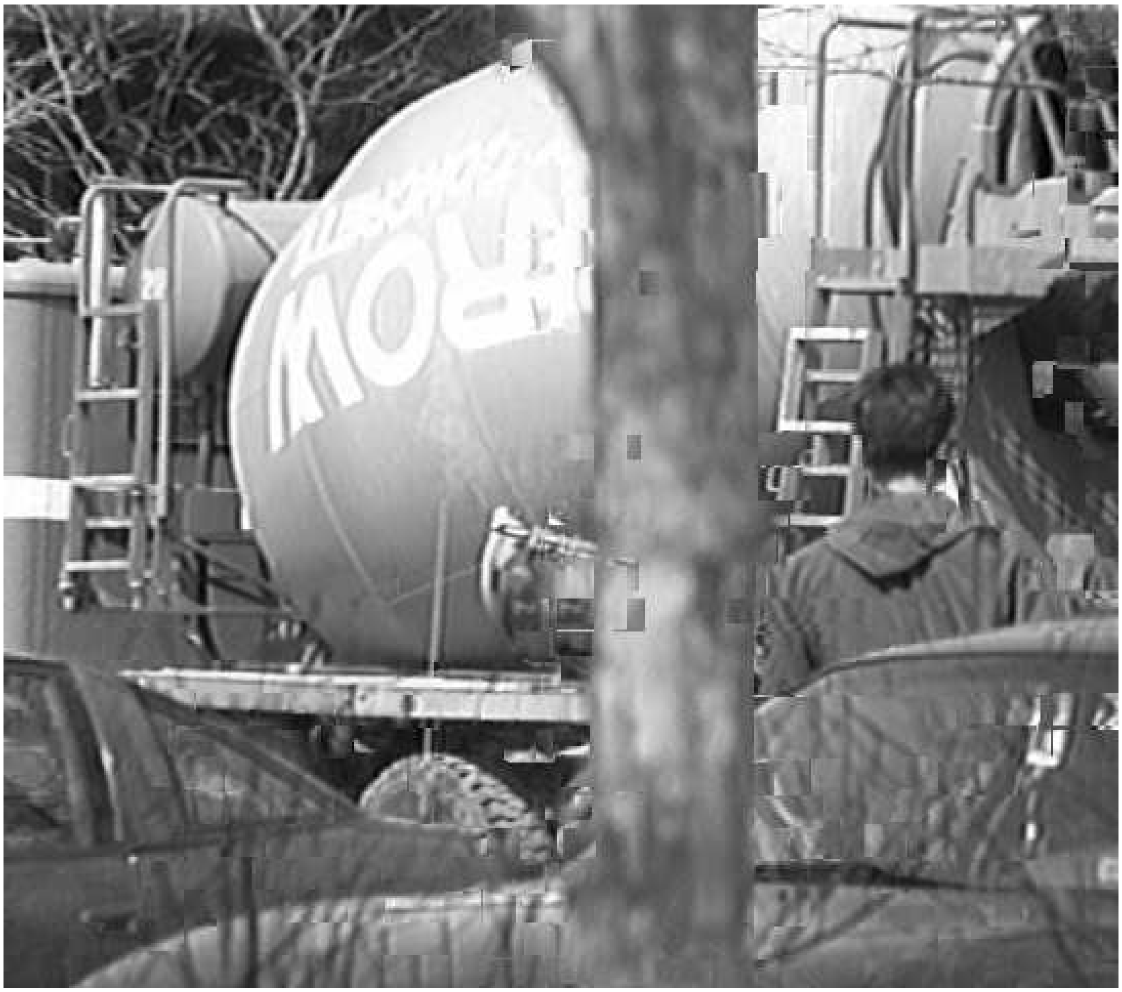


Figure 6.22: *Cement Mixer* $F(1.5,175)$ interpolated without displacement field processing.

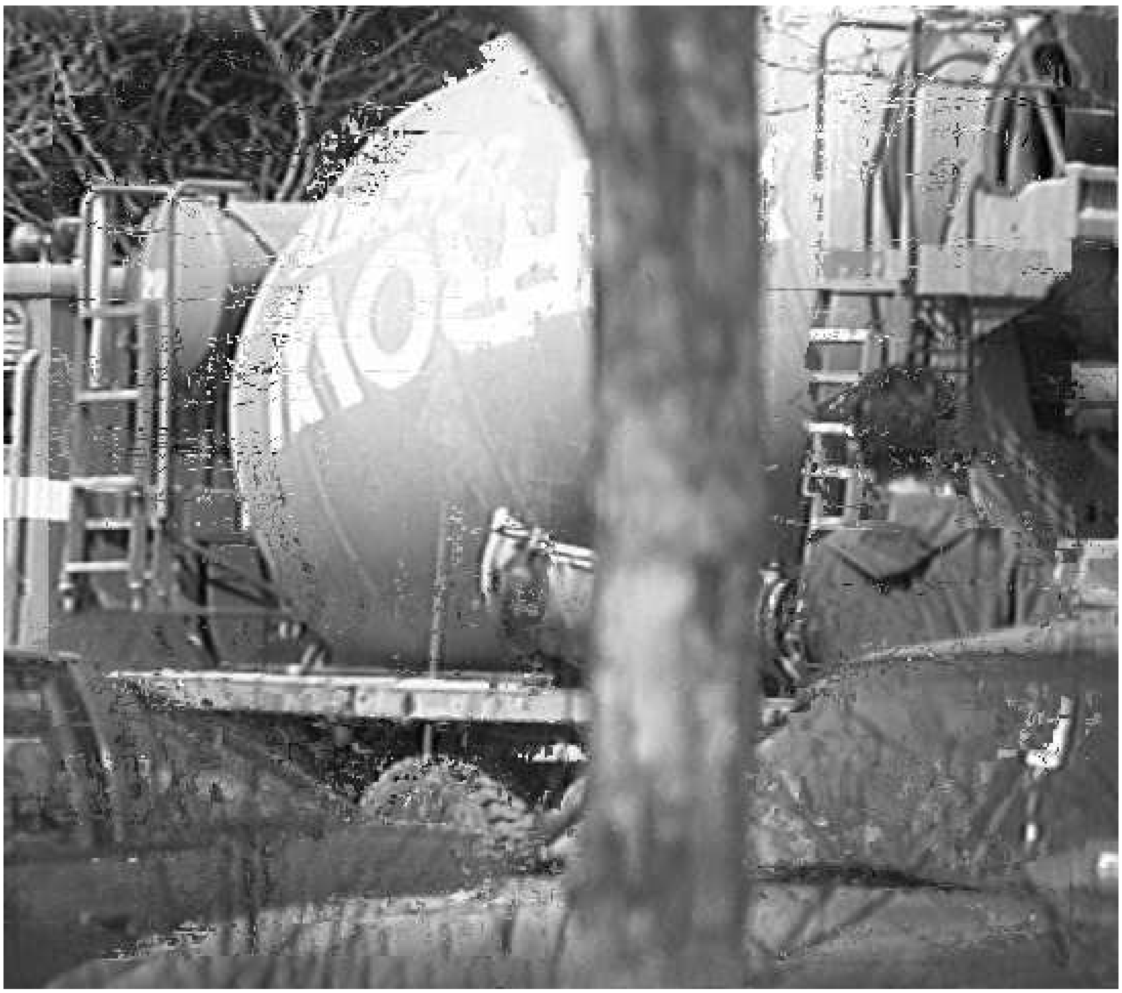


Figure 6.23: *Cement Mixer F*(1.5,175) interpolated using range ordering for ambiguously-referenced image regions and zero displacement occlusion filling.

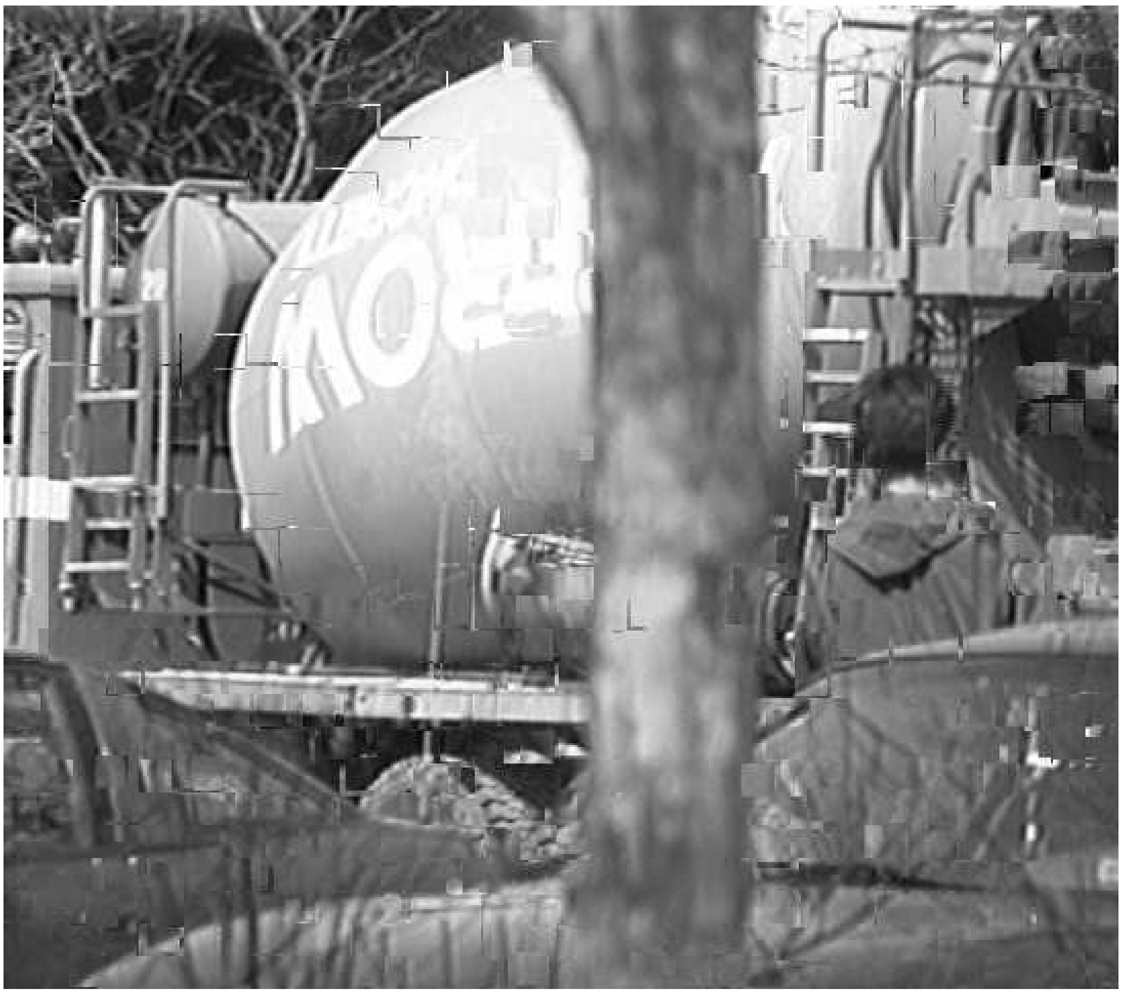


Figure 6.24: *Cement Mixer F*(1.5,175) interpolated using unprocessed field, range ordering for the selection of ambiguously-referenced image regions, and filling of holes with zero displacement.

a.

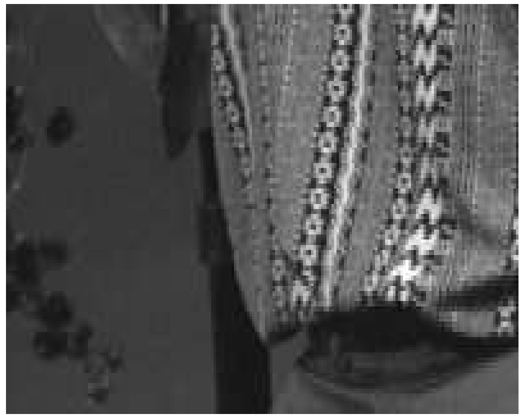
b.

Figure 6.25: *Piano*, (a) original $F(0,25)$, (b) original $F(1,25)$

a.

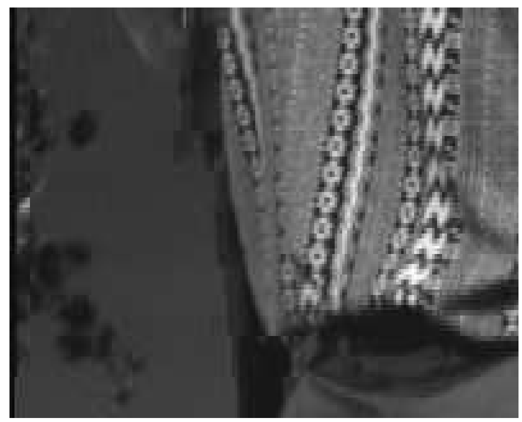
b.

Figure 6.26: *Piano F*(0.5,25), (a) interpolated using improved technique, (b) interpolated using basic scheme.

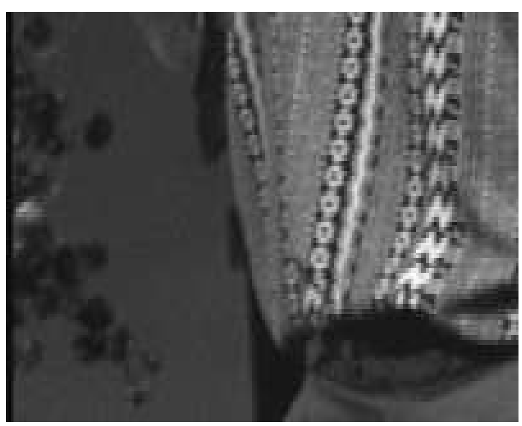


a.

b.



c.



d.

Figure 6.27: Close-ups of *Piano* frame, (a) original $F(0,25)$, (b) interpolated $F(0.5,25)$ using improved technique (c) interpolated $F(0.5,25)$ using basic interpolation scheme, (d) original $F(1,25)$

Chapter 7

Conclusions and Future Work

In this thesis, we addressed the problem of efficiently compressing multiple, concurrent views of a scene, obtained from cameras arranged in an arbitrary (non-periodic) configuration. The underlying concept behind this work is that single-view video signal characteristics, typically exploited by coding algorithms, do not necessarily apply to multi-view signals. For example, in these signals: 1) the relative location of the optimal reference frame depends on the structure and motion of both scene objects and cameras, 2) significant correlation exists between the predicted and residual images due to lighting and camera pick-up variations, and 3) most scene objects possess non-zero displacement between frames within separate views. We developed three distinct algorithms that utilized the special characteristics of multi-view signals with the goal of achieving superior rate-distortion performance while maintaining acceptable complexity and storage costs.

Improved frame-based prediction was achieved by a simple method to adaptively select the best possible reference for the frame to be encoded. The selection was based on the adverse effect of occlusions on prediction performance, and the one-to-one relationship between occlusions and discontinuities within a displacement vector field. For constant quality reconstructed images, this technique realized a bit rate reduction of 10-30% when compared with fixed, pre-selected reference frame schemes; it also achieved rate-distortion performance within 1% of an exhaustive search method, while consuming considerably fewer processing cycles. These gains were offset by a storage requirement approximately 2-3 times that of the fixed or exhaustive search methods.

A novel class of residual encoders was presented that is capable of exploiting correlation between the predicted and residual images. The functionality of this coder resembles that of a predictive vector quantizer, with the significant difference that a unique vector predictor is associated with each reproduction vector. This explicit coupling allows for the tracking of non-stationarities in the source signal without the transmission of side information; it also provides for the joint optimization of the predictor and quantizer. For fixed average distortion, this technique required roughly 15% fewer bits compared with a comparable vector quantizer; for a constant bit rate of approximately 0.13 bpp, the average gain in signal-to-noise ratio was over 0.8 dB. Unfortunately, these gains were obtained at the expense of high encoder complexity and storage costs.

An enhanced viewpoint interpolation scheme was developed, which requires the use of only two decoded exterior views and the noisy, block-based, displacement vector field extracted from the coded bit stream relating these views. Problems typically encountered in interpolation schemes were minimized by: rectifying displacement estimation errors, due to displacement discontinuities within an image block, via the self-synthesis procedure; inferring range information for occlusions from neighboring unoccluded regions; and selecting the correct displacement for ambiguously-referenced regions by using a displacement estimate confidence measure. This technique was applied to numerous stereo-pairs, and subjectively-pleasing intermediate views were obtained; impressive results were realized even for image pairs with relatively large displacement ranges.

A summary of the relative costs for our novel reference frame selection, residual encoding, and viewpoint interpolation algorithms are provided in Tables 7.1, 7.2, and 7.3, respectively. The cost values were obtained from the simulations performed on our set of multi-view signals, and they were normalized with respect to the corresponding baseline technique. Definitions of these costs are provided in Section 2.3, where we used the mean-squared-error distortion function, and measured complexity in terms of the processing time to perform each technique.

Figure 7.1 illustrates a complete multi-view hybrid coder structure that incorporates all three of these algorithms. We observe that any or all of these techniques may be employed. With respect to overall system cost, we reach the following conclusions. If encoder storage is not a concern, adaptive reference frame selection should be performed – the rate-distortion performance afforded by this technique far outweighs it relatively moderate complexity cost. Only in systems that have severe bit rate constraints should the restoration-based coder be utilized – more substantial improvements will need to be observed, on diverse multi-view signals, for this technique to overcome its complexity and storage requirements. Finally, due to its superior performance and minimal costs, interior views should not be generated, and, instead, viewpoint interpolation should be performed at the decoder.

Besides the future work described for each algorithm in their respective chapter summaries, we plan on applying these techniques to more elaborate multi-view signals, i.e., number of views greater than two. The techniques that prove most useful then should be implemented into a real-time, multi-view communication system possessing the desirable features of simulated motion parallax and viewer-specified level of stereoscopy.

Technique	Rate	Distortion	Complexity	Storage
Fixed	1.00	1.00	1.00	1.00
Exhaustive	0.79	1.00	3.83	1.00
Adaptive	0.78	1.00	2.40	2.06

Table 7.1: Relative encoder cost comparison summary between fixed, exhaustive, and adaptive reference frame selection schemes.

Technique	Rate	Distortion	Complexity	Storage
PTSVQ	1.00	1.00	1.00	1.00
PTSVR	0.87	0.99	3.12	1.37

Table 7.2: Relative encoder cost comparison summary between vector quantization (PTSVQ) and vector restoration (PTSVR) residual encoder implementations.

Technique	Rate	Distortion	Complexity	Storage
Basic	1.00	1.00	1.00	1.00
Enhanced	1.00	0.58	1.32	1.39

Table 7.3: Relative decoder cost comparison summary between basic and enhanced viewpoint interpolation techniques.

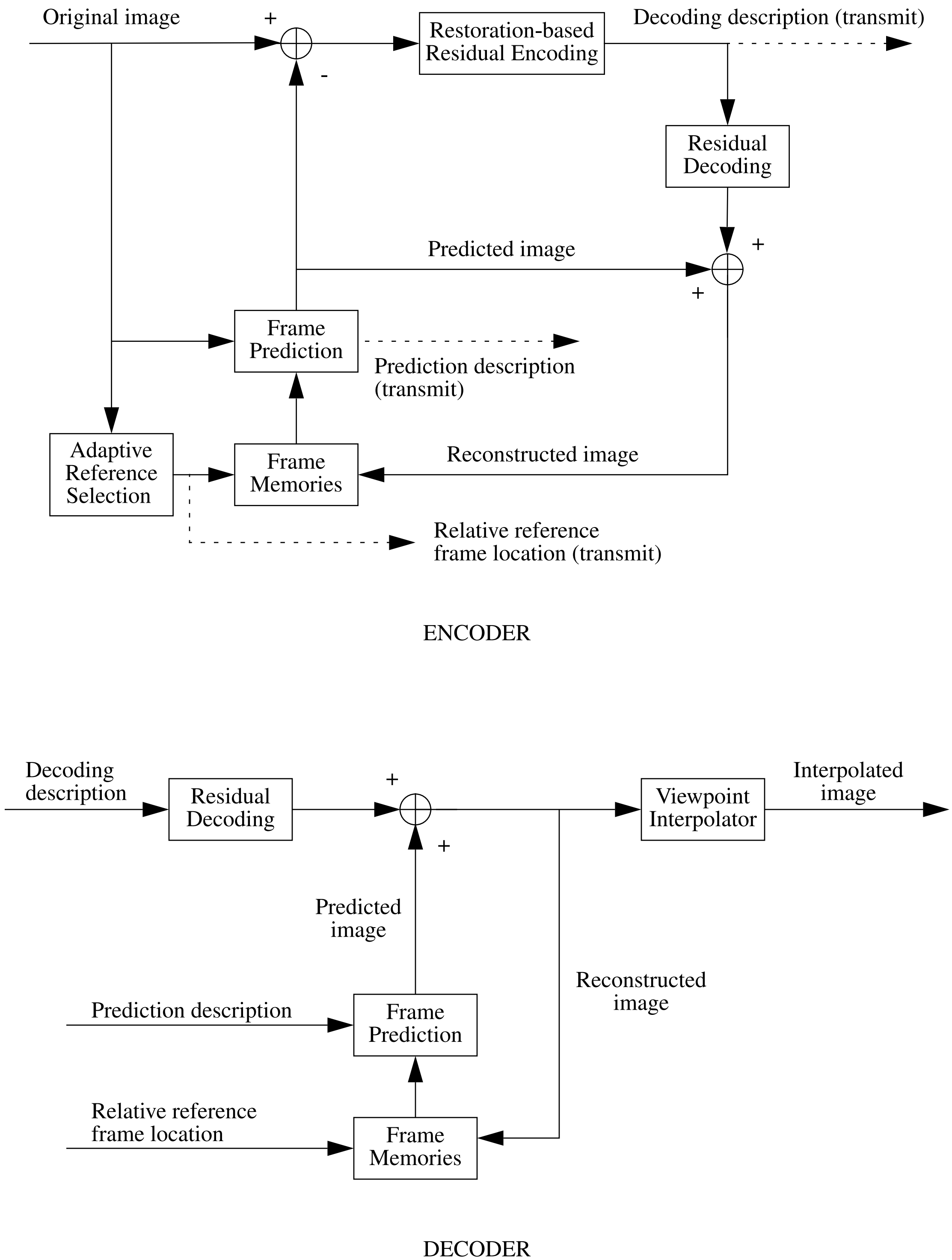


Figure 7.1: Complete multi-view hybrid coder structure incorporating adaptive reference frame selection, restoration-based residual coding, and viewpoint interpolation from a noisy displacement vector field.

Bibliography

[1] CCITT Study Group XV - Report R95, Recommendation H.261, "Video Codec for Audiovisual Services at p x 64 kbits," May 1992.

[2] ISO/IEC JTC1/SG29/WG11, ISO/IEC 11172-2, "Information Technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbits/s - Part 2: Video", May 1993.

[3] ISO/IEC JTC1/SC29/WG11 Test Model Editing Committee, "MPEG-2 Video Test Model 5", ISO/IEC JTC1/SC29/WG11 Doc. N0400, April 1993.

[4] ISO/IEC JTC1/SC29/WG11, "Information Technology - Generic Coding of Moving Pictures and Associated Audio, Recommendation H.262, ISO/IEC 13818-2, Draft International Standard," March 1994.

[5] G. P. Abousleman, M. W. Marcellin and B. R. Hunt, "Compression of hyperspectral imagery using the 3-D DCT and hybrid DPCM/DCT," *IEEE Trans. Geoscience and Remote Sensing*, vol. 33, no. 1, pp. 26-34, Jan. 1995.

[6] 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, Feb. 1996.

[7] H. Aydinoglu and M. H. Hayes, "Compression of multi-view images," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 385-389, Nov. 1994.

[8] N. Balram and J. M. F. Moura, "Noncausal Gauss Markov random fields: Parameter structure and estimation," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 334-354, Mar. 1993.

[9] H. H. Baker and R. C. Bolles, "Generalizing epipolar plane image analysis on the spatiotemporal surface," *Int. J. of Computer Vision*, vol. 3, no. 1, pp. 33-49, 1989.

[10] S. T. Barnar and M. A. Fischler, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 2, pp. 333-340, July 1980.

[11] R. C. Bolles, H. H. Baker and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *Int. J. of Computer Vision*, vol. 1, no. 1, pp. 176-185, 1987.

[12] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees. The Wadsworth Statistics / Probability Series*, Belmont CA: Wadsworth, 1984.

[13] C. Cafforio, F. Rocca and S. Tubaro, "Motion compensated image interpolation," *IEEE Trans. Comm.*, vol. 38, no. 2, pp. 215-222, Feb. 1990.

[14] E. Chalom, E. and V. M. Bove, V.M, "Segmentation of frames in a video sequence using motion and other attributes," in *Proc. Digital Video Compression: Algorithms and Technologies 1995*, SPIE, vol. 2419, pp. 230-241, 1995.

[15] R. Chassaing, B. Choquet and D. Pele, "A stereoscopic television system (3D-TV) and compatible transmission on a MAC channel (3D-MAC)", *Signal Processing: Image Communication*, vol. 4, no. 1, pp. 33-43, Nov. 1991.

[16] P. A. Chou, T. Lookabaugh and R. M. Gray, "Optimal pruning with application to tree-structured source coding and modeling," *IEEE Tran. Inform. Theory*, vol. 35, no. 2, pp. 299-315, 1989.

[17] P. A. Chou, T. Lookabaugh and R. M. Gray, "Entropy-constrained vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 1, pp. 31-42, 1989.

[18] R. J. Clarke, *Transform Coding of Images*, London: Academic Press, 1985.

[19] L. Corte-Real and A. P. Alves, "A very low bit rate video coder based on vector quantization," *IEEE Trans. Image Processing*, vol. 5, no. 2, pp. 263-273, Feb. 1996.

[20] J. L. Crowley, P. Bobet and C. Schmid, "Auto-calibration by direct observation of objects," *Image and Vision Computing*, vol. 11, no. 2, pp. 67-81, Mar. 1993.

[21] G. Demoment, "Image reconstruction and restoration: Overview of common estimation structures and problems," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 2024-2036, Dec. 1989.

[22] I. Dinstein, M. G. Kim, A. Henik and J. Tzelgov, "Compression of stereo images using subsampling and transform coding," *Optical Engineering*, vol. 30, no. 9, pp. 1359-1364, Sept. 1991.

[23] I. Dinstein, G. Guy, J. Rabany, J. Tzelgov and A. Henik, "On the compression of stereo images: Preliminary results," *Signal Processing*, vol. 17, no. 4, pp. 373-382, Aug. 1989.

[24] M. Effros and P. A. Chou, "Weighted universal transform coding: Universal image compression with the Karhunen-Loève Transform," in *Proc. IEEE Internat. Conf. Image Processing*, vol. 2, pp. 61-64, 1995.

[25] D. Eggert and K. Bowyer, "Computing the perspective projection aspect graph of solids of revolution," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 15, no. 2, pp. 109-128, 1993.

[26] N. Farvardin and J. W. Modestino, "Rate-distortion performance of DPCM schemes for autoregressive sources," *IEEE Trans. Inform. Theory*, vol., 31, no. 3, pp. 402-418, May 1985.

[27] J. D. Fooley, *Introduction to Computer Graphics,* Reading, MA: Addison-Wesley, 1994.

[28] M. Foodeei and E. Dubois, “Coding image sequence intensities along motion trajectories using EC-CELP quantization,” in *Proc. IEEE Int. Conf. Image Processing*, pp. 720-724, 1994.

[29] M. Foodeei and E. Dubois, “Rate-distortion performance of source coders in the low bit-rate region for highly correlated Gauss-Markov source,” in *Proc. GLOBECOM*, pp. 123-127, 1993.

[30] T. Fujii and H. Harashima, “Data compression of an autostereoscopic 3-D image,” University of Tokyo Tech. Rep., 1994.

[31] B. Gerod, “The efficiency of motion-compensating prediction for hybrid coding of video sequences,” *IEEE J. Select. Areas Commun*., vol. 5, pp. 1140-1154, Aug. 1987

[32] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Boston: Kluwer Academic, 1991.

[33] V. S. Grinberg, G. Podnar and M. W. Siegel, “Geometry of binocular imaging,” in *Proc. Stereoscopic Displays and Virtual Reality Systems*, SPIE, vol. 2177, pp. 56-65, 1994.

[34] E. Grosso and M. Tistarelli, “Active/dynamic stereo vision,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 17, no. 9, pp. 868-879, Sept. 1995.

[35] S. Gupta and A. Gersho, “Feature predictive vector quantization of multispectral images,” *IEEE Trans. Geoscience and Remote Sensing*, vol. 30, no. 3, pp. 491-501, May 1992.

[36] S. Gupta and A. Gersho, “Nonlinear predictive vector quantization of multispectral imagery,” in *Proc. Twenty-Fourth Asilomar Conf. Signals, Systems and Computers*, pp. 331-335, 1990.

[37] C. Hansen, N. Ayache and F. Lustman, “Efficient depth estimation using trinocular stereo,” in *Proc. Sensor Fusion: Spatial Reasoning and Scene Interpretation*, Cambridge, MA, SPIE, vol. 1003, pp. 124-131, 1988.

[38] R. Hopkins, “Digital terrestrial HDTV for North America: The Grand Alliance HDTV system,”, *IEEE Trans. Consumer Electronics*, vol. 40, no. 3, pp. 185-198, Aug. 1994.

[39] C.-H. Hsieh and J.-S. Shue, “Frame adaptive finite-state vector quantization for image sequence coding,” *Signal Processing: Image Communication*, vol. 7, pp. 13-26, 1995.

[40] R. Hsu, K. Kodama and H. Harashima, “View interpolation using epipolar plane images,” in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 745-749, 1994.

[41] C.-L. Huang and T.-T. Chao, “Motion-compensated interpolation for scan rate up-conversion,” *Optical Engineering*, vol. 35, no. 1, pp. 166-176, Jan. 1996.

[42] A. K. Jain, *Fundamentals of Digital Image Processing*, Englewood Cliffs NJ: Prentice Hall, 1989.

[43] A. K. Jain, "Advances in mathematical models for image processing," *Proc. IEEE*, vol. 69, no. 5, pp. 502-528, May 1981.

[44] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Comm.*, vol. 29, pp. 1799-1808, Dec. 1981.

[45] B. Julesz, *Foundations of Cyclopean Perception*, Chicago: The University of Chicago Press, 1971.

[46] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," in *Proc. Int. Conf. Robotics and Automation*, pp. 1088-1095, 1991.

[47] A. Katayama, K. Tanaka, T. Oshino and H. Tamura, "A viewpoint dependent stereoscopic display using interpolation of multi-viewpoint images," in *Proc. Stereoscopic Displays and Virtual Reality Systems II*, SPIE, vol. 2409, pp. 11-20, 1995.

[48] J.-S. Kim and R.-H. Park, "Local motion-adaptive interpolation technique based on block matching algorithms," *Signal Processing: Image Communications*, vol. 4, no. 6, pp. 519-528, Nov. 1992.

[49] Y. Kim, I. Choi, I. Lee, T. Yun, and K. T. Park, "Wavelet transform image compression using human visual characteristics and a tree structure with a height attribute," *Optical Engineering*, vol. 35, no. 1, pp. 204-212, Jan. 1996.

[50] Y. H. Kim and J. W. Modestino, "Adaptive entropy-coded pruned tree-structured predictive vector quantization of images," *IEEE Trans. Comm.*, vol. 41, no. 1, Jan. 1993.

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

[52] E. Krotkov, M. Hebert and Simmons, "Stereo perception and dead reckoning for a prototype lunar rover," *Autonomous Robots*, vol. 2, no. 4, pp. 313-331, 1995.

[53] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Processing Mag.*, vol. 13, no. 3, pp. 43-64, May 1996.

[54] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantization design," *IEEE Tran. Comm.*, vol. 28, no. 1, pp. 84-95, 1980.

[55] L. Lipton, *The CrystalEyes Handbook*, San Rafael CA: StereoGraphics Corporation, 1991.

[56] J. Liu and R. Skerjanc, "Stereo and motion correspondence in a sequence of stereo images," *Signal Processing: Image Communication*, vol. 5, pp. 305-318, 1993.

[57] C. Loeffler, A. Ligtenberg and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, pp. 988-991, 1989.

[58] T. Lookabaugh, E. A. Riskin, P. A. Chou and R. M. Gray, "Variable rate vector quantization for speech, image, and video compression," *IEEE Trans. Comm.*, vol. 41, no. 1, pp. 186-199, Jan. 1993.

[59] M. E. Lukacs, "Predictive coding of multi-viewpoint image sets," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 521-524, 1986.

[60] J. N. Mailhot and H. Derovanessian, "Grand Alliance HDTV video encoder," *IEEE Trans. Consumer Electronics*, vol. 41, no. 4, pp. 1014-1019, Nov. 1995.

[61] F. C. M. Martins and J. M. F. Moura, "3-D video compositing: Towards a compact representation for video sequences," in *Proc. IEEE Int. Conf. Image Processing*, pp. 550-553, 1995.

[62] T. Miyashita and T. Uchida, "Cause of fatigue and its improvement in stereoscopic displays," *Proc. SID*, vol. 31, no. 3, pp. 249-254, 1990.

[63] J. S. McVeigh, M. W. Siegel and A. G. Jordan, "Intermediate view synthesis considering occluded and ambiguously referenced image regions," *Signal Processing: Image Communications*, accepted.

[64] J. S. McVeigh, M. W. Siegel and A. G. Jordan, "Adaptive reference frame selection for generalized video signal coding," in *Proc. Digital Video Compression: Algorithms and Technologies 1996*, SPIE, vol. 2668, pp. 441-449, 1996.

[65] J. S. McVeigh, M. W. Siegel and A. G. Jordan, "Algorithm for automated eye-strain reduction in real stereoscopic images and sequences," in *Proc. Human Vision and Electronic Imaging*, SPIE, vol. 2657, pp. 307-316, 1996.

[66] J. S. McVeigh, S.-W. Wu, M. W. Siegel and A. G. Jordan, "Vector restoration for video coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, pp. 93-96, 1995.

[67] J. S. McVeigh, V. S. Grinberg and M. W. Siegel, "Double buffering technique for binocular imaging in a window," in *Proc. Stereoscopic Displays and Virtual Reality Systems II*, SPIE, vol. 2409, pp. 168-175, 1995.

[68] J. S. McVeigh and S.-W. Wu, "Partial closed loop versus open loop motion estimation for HDTV compression," *Int. J. Imaging Systems and Technology*, vol. 5, no. 4, pp. 268-275, 1994.

[69] N. M. Nasrabadi, C. Y. Choo and Y. S. Feng, "Finite-state vector quantization of digital images," *IEEE Trans. Comm.*, vol. 32, no. 5, pp. 2145-2154, May 1994.

[70] A. Ortega and M. Vetterli, "Adaptive quantization without side information," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 856-860, 1994.

[71] K. K. Pang and T. K. Tan, "Optimum loop filter in hybrid coders," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 4, no. 2, pp. 158-167, Apr. 1994.

[72] S. Panis and M. Ziegler, "Object-based coding using motion and stereo information," in *Proc. Picture Coding Symposium*, pp. 308-312, 1994.

[73] A. Papoulis, *Probability, random variables, and stochastic processes*, New York: McGraw-Hill, 1991, ch. 12.

[74] S. Pastoor, "3D-television: A survey of recent research results on subjective requirements," *Signal Processing: Image Communication*, vol. 4, no. 1, pp. 21-32, Nov. 1991.

[75] S. Pastoor and K. Schenke, "Subjective assessments of the resolution of viewing directions in a multi-viewpoint 3D TV system," *Proc. SID*, vol. 30, no. 3, pp. 217-223, 1989.

[76] M. G. Perkins, "Data compression of stereopairs," *IEEE Trans. Comm*, vol. 40, no. 4, pp. 684-696, Apr. 1992.

[77] A. Puri and B. Haskell, "Straw man proposal for multi-view profile," ISO/IEC JTC1/SC29/WG11 MPEG95/485, Nov. 1995.

[78] A. Puri, R. V. Kollarits and B. G. Haskell, "Stereoscopic video compression using temporal scalability," in *Proc. Visual Communications and Image Processing*, SPIE, vol. 2501, pp. 745-756, 1995.

[79] R. L. de Queiroz and K. R. Rao, "Variable block size lapped transforms," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 290-293, 1995.

[80] K. R. Rao and P. Yip, *Discrete Cosine Transform - Algorithms, Advantages, Applications*, London: Academic Press, 1990, appendix A.2.

[81] C. Reader, "MPEG4: coding for content, interactively, and universal accessibility," *Optical Engineering*, vol. 35, no. 1, pp. 104-108, Jan. 1996.

[82] J. Ribas-Corbera and D. L. Neuhoff, "Optimal bit allocations for lossless video coders: Motion vectors vs. difference frames," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 180-183, 1995.

[83] J. Ribas-Corbera and J. Sklansky, "Interframe interpolation of cinematic sequences," *J. Visual Communication and Image Representation*, vol. 4, no. 4, pp. 392-406, Dec. 1993.

[84] P. Richardson, "Image restoration using vector classified adaptive filtering," in *Proc. Visual Communication and Image Processing*, SPIE, vol. 2094, pp. 1581-1591, 1993.

[85] S. A. Rizvi and N. M. Nasrabadi, "Predictive residual vector quantization," *IEEE Trans. Image Processing*, vol. 4, no. 11, pp. 1482-1495, Nov. 1995.

[86] A. Schertz, "Source coding of stereoscopic television pictures", in *Proc. IEE Int. Conf. Image Processing and its Applications*, pp. 462-464, 1992.

[87] S. Sethuraman, M. W. Siegel, and A. G. Jordan, "A multiresolution framework for stereoscopic image sequence compression," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, pp. 361-365, 1994.

[88] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criteria," in *IRE Nat. Conv. Rec.*, vol. 4, pp. 142-163, 1959.

[89] R. Skerjanc and J. Liu, “A three camera approach for calculating disparity and synthesizing intermediate pictures,” *Signal Processing: Image Communication*, vol. 4, no. 1, pp. 55-64, Nov. 1991.

[90] P. W. Smith and N. Nandhakumar, “An improved power cepstrum based stereo correspondence method for textured scenes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 3, pp. 338-348, Mar. 1996.

[91] J. Takeno and U. Rembold, “Stereovision systems for autonomous mobile robots,” in *Proc. Int. Conf. Intelligent Autonomous Systems*, pp. 26-41, 1995.

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

[93] R. Thoma and M. Bierling, “Motion compensating interpolation considering covered and uncovered background,” *Signal Processing: Image Communications*, vol. 1, no. 2, pp. 191-212, Oct. 1989.

[94] D. Tzovaras, N. Grammalidis, M. G. Strintzis, “Joint three-dimensional motion/disparity segmentation for object-based stereo image sequence coding,” *Optical Engineering*, vol. 35, no. 1, pp. 137-144, Jan. 1996.

[95] N. A. Valyus, *Stereoscopy*, London: The Focal Press, 1966.

[96] H. Van Trees, *Detection, Estimation, and Modulation Theory,* New York: Wiley, 1968, vol. 1, ch. 2.

[97] C. Ware, C. Gobrecht and M. Paton, “Algorithm for dynamic disparity adjustment,” in *Proc. Stereoscopic Displays and Virtual Reality Systems II*, SPIE, vol. 2409, pp. 150-156, 1995.

[98] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series*, Cambridge MA: MIT Press, 1949.

[99] C. D. Wilkens, “Three-dimensional stereoscopic display implementation: Guidelines derived from human visual capabilities,” in *Proc. Stereoscopic Displays and Applications*, SPIE, vol. 1256, pp. 2-11, 1990.

[100] X. Wu and Y. Fang, “A segmentation-based predictive multiresolution image coder,” *IEEE Trans. Image Processing*, vol. 4, no. 1, pp. 34-47, Jan. 1995.

[101] H. Yamaguchi, “Multifocus synthesis and its application to 3-D image capturing,” in *Proc. Visual Communications and Image Processing*, SPIE, vol. 2094, pp. 281-291, 1993.

[102] H. Yamaguchi, Y. Tatehira, K. Akiyama and Y. Kobayashi, “Stereoscopic images disparity for predictive coding,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 1976-1979, 1989.

[103] T. Yamazaki, K. Kamijo and S. Fukuzumi, “Quantitative evaluation of visual fatigue encountered in viewing stereoscopic 3D displays: Near-point distance and visual evoked potential study,” *Proc. SID*, vol. 31, no. 3, pp. 245-247, 1990.

[104] A. Zakhor and F. Lari, "Edge-based 3-D camera motion estimation with application to video coding," *IEEE Trans. Image Processing*, vol. 2, no. 4, pp. 481-498, Oct. 1993.

[105] K. Zeger and A. Bist, "Universal source coding with codebook transmission," *IEEE Trans. Comm.*, vol. 42, no. 2-4, pp. 336-346, 1994.

[106] K. Zeger and A. Bist, "Universal adaptive vector quantization using codebook quantization with application to image compression," in *Proc. Int. Conf. Acoust., Speech, Signal Processing*, vol. 5, pp. 381-384, 1992.

[107] Q. Zheng and R. Chellappa, "A computational vision approach to image registration," *IEEE Trans. Image Processing*, vol. 2, no. 3, pp. 311-325, July 1993.

[108] W. Zschunke, "DPCM picture coding with adaptive prediction," *IEEE Trans. Comm.*, vol. 25, no. 11, pp. 1295-1302, Nov. 1977.