Adaptive Format Conversion Information as Enhancement Data for Scalable Video Coding

by

Wade K. Wan

E.E., Massachusetts Institute of Technology (2001)
S.M., Massachusetts Institute of Technology (1998)
B.S., Johns Hopkins University (1995)

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2002

Author_____
Department of Electrical Engineering and Computer Science
May 24, 2002

Certified by_____
Jae S. Lim
Professor of Electrical Engineering
Thesis Supervisor

Accepted by_____
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

Adaptive Format Conversion Information as Enhancement Data for Scalable Video Coding

by

Wade K. Wan

Submitted to the Department of Electrical Engineering and Computer Science
on May 24, 2002, in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering

Abstract

Scalable coding techniques can be used to efficiently provide multicast video service and involve transmitting a single independently coded base layer and one or more dependently coded enhancement layers. Clients can decode the base layer bitstream and none, some or all of the enhancement layer bitstreams to obtain video quality commensurate with their available resources. In many scalable coding algorithms, residual coding information is the only type of data that is coded in the enhancement layers. However, since the transmitter has access to the original sequence, it can adaptively select different format conversion methods for different regions in an intelligent manner. This adaptive format conversion information can then be transmitted as enhancement data to assist processing at the decoder. The use of adaptive format conversion has not been studied in detail and this thesis examines when and how it can be used for scalable video compression.

A new scalable codec is developed in this thesis that can utilize adaptive format conversion information and/or residual coding information as enhancement data. This codec was used in various simulations to investigate different aspects of adaptive format conversion such as the effect of the base layer, a comparison of adaptive format conversion and residual coding, and the use of both adaptive format conversion and residual coding. The experimental results show adaptive format conversion can provide video scalability at low enhancement bitrates not possible with residual coding and also assist residual coding at higher enhancement layer bitrates. This thesis also discusses the application of adaptive format conversion to the migration path for digital television. Adaptive format conversion is well-suited to the unique problems of the migration path and can provide initial video scalability as well as assist a future migration path.

Thesis Supervisor: Jae S. Lim
Title: Professor of Electrical Engineering

Dedication

To

Mom and Dad

Acknowledgements

There are many people who have helped make this thesis possible. I have learned a great deal from them during my graduate career and would like to acknowledge their contributions.

First, I would like to thank my thesis supervisor Professor Jae Lim for providing me with the opportunity to work in his group and the intellectual and financial support necessary to complete my thesis. I would also like to thank Professor David Staelin and Dr. Dan Dudgeon for serving on my thesis committee and providing useful comments that improved this thesis. I would like to acknowledge both the Advanced Television Research Program (ATRP) and the Intel Foundation for their financial support of my thesis research. Thanks to Professor Roger Mark for being my academic advisor.

I have been fortunate to work with the members of the ATRP and my interactions with them have been rewarding both professionally and personally. I don't know how I can ever properly thank Cindy LeBlanc, the group administrative assistant, for everything she has done to make my life easier. She has always been willing to provide me assistance, has constantly looked out for my well-being and been a supportive friend from my first day in the lab. I am grateful to my fellow Ph.D. students Eric Reed, Ray Hinds and David Baylon who helped me get started when I first joined the lab and continued to assist me both during and after their careers at MIT. I would like to thank Shiufun Cheung for providing the thesis template I used to format this thesis. Thanks to James Thornbrue for reviewing a manuscript of this thesis and providing valuable comments. Special thanks to Brian Heng for being both a valuable colleague as well as a close friend.

I would like to thank Ajay Luthra, Xuemin Chen and Krit Panuspone for the opportunity to work with them during my summer internships at General Instrument Corporation. I learned a great deal about the video processing field from these internships and this knowledge has been invaluable for my thesis research.

I am extremely fortunate to have wonderful friends who have been very supportive throughout my graduate education. I am very grateful to Ramakrishna Mukkamala who I met early in my graduate career and has been a trusted friend who has always been there for me. I also would like to thank Everest Huang, Jeff Craig, Kip Chaney, David Ettenson and Chuck Cheung for their friendship and support through both good and difficult times.

Finally, I would like to thank my parents, Wing-Yuk and Manwai, and my brother Jimmy. My family has always been an source of unending love, encouragement, support and patience. I am extremely privileged to have the opportunity to work towards a Ph.D. and thank my parents for providing me with all of the opportunities that led up to and include my graduate studies.

Wade Wan
Cambridge, MA
May 24, 2002

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Multicast Video Service

Many video broadcasting applications must provide service to a *multicast* environment. In this environment, multiple client types require different types of service, also known as service levels, due to the variations in bandwidth, processing power and memory resources available to each client type. A video server would like to provide different resolutions and/or qualities of the same video sequence to satisfy each client type while minimizing the cost of reaching the audience. From the video coding point of view, this cost is the total bandwidth transmitted across the network from the server. There may be other issues to consider depending on the specific application such as codec complexity.[1] However, this thesis will consider these issues to be secondary in the analysis and use the total transmitted bandwidth as the only measure of cost. Figure 1.1 shows an example of a multicast environment with three client types. In this example, each client type is connected to the network by a different communication link (56k modem, cable modem and LAN (local area network)). Therefore, each client type requires a different type of video service due to the differences in their connectivity (and available bandwidth) to the network.

Scalable coding techniques [1] can be used to efficiently achieve multiple levels of video service and involve transmitting a single independently coded base layer and one or more dependently coded enhancement layers. Each enhancement layer is dependent on the base layer as well as the previous enhancement layer(s) and increases the resolution or the quality of the decoded video compared to the previous layer. Clients can decode the base layer bitstream and none, some or all of the enhancement layer bitstreams to obtain

[1] A codec is defined to be an encoder and its corresponding decoder.

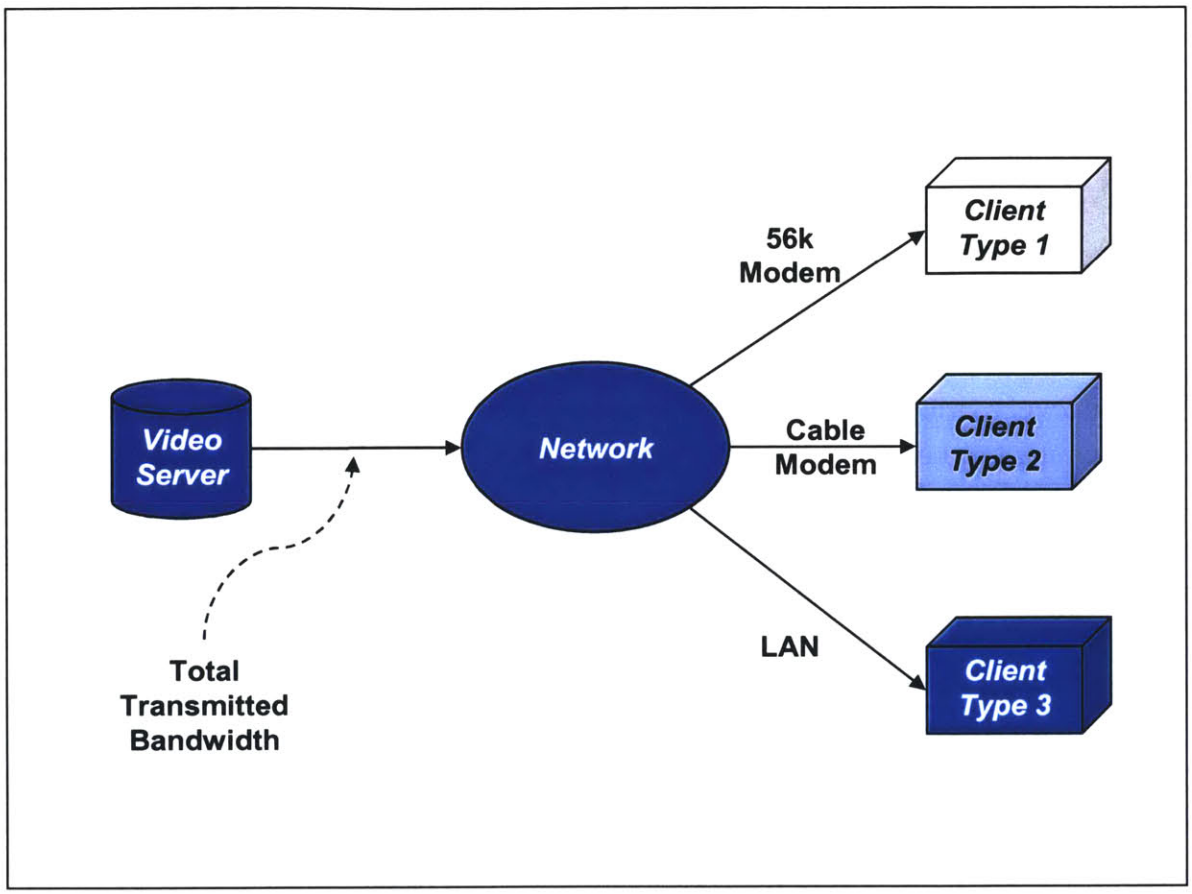

Figure 1.1: Example of a Multicast Environment. A single video server would like to provide content to multiple client types that require different service levels. In this example, the three client types require different video service levels since each client type is connected to the network with a different communication link (56k modem, cable modem and LAN). One measure of cost is the total bandwidth that must be transmitted across the network by the server to provide multicast service.

video quality commensurate with their available resources. The multicast environment shown in Figure 1.1 can be satisfied using a scalable coding scheme with three layers (a base layer and two enhancement layers) as shown in Figure 1.2. Clients with a 56k modem would decode only the base layer and receive "good" video quality. The first enhancement layer could be decoded in addition to the base layer by clients with a cable modem to receive "better" video quality. LAN clients could decode the base layer and both enhancement layers to receive the "best" video quality.

1.2 Motivation for Thesis

The efficient coding of information in the enhancement layers of scalable coding schemes is an active research area. Each enhancement layer in a scalable compression scheme can increase the resolution or quality of the decoded video. Consider a scalable coding scheme where each enhancement layer increases the resolution of the decoded video. The additional resolution of enhancement layers is first provided by a format conversion of a previously decoded lower resolution format to a higher resolution format. The *residual* is defined to be the difference between this sequence and the original high resolution sequence. This "error" signal is often coded and added to the decoded sequence to improve its quality. Therefore, there are two types of information that can be utilized in a scalable coding scheme: *residual coding* and *adaptive format conversion*. Residual coding is well understood and used in many scalable coding algorithms such as the spatial scalability profiles in the MPEG-2 and MPEG-4 multimedia coding standards. [2, 3, 4] The use of adaptive format conversion as enhancement data is not as well studied and often overlooked since many scalable coding schemes use a fixed method of format conversion for the entire video sequence and only utilize residual coding information for enhancement data. However, since the transmitter has access to the original high resolution sequence, it can adaptively select different format conversion methods for different regions in an intelligent manner. This will often result in much better performance than nonadaptive format conversion since one format conversion method usually does not work well for the entire sequence. [5, 6] The fundamental motivation for this thesis is to determine when and how adaptive format conversion can be used to improve scalable video compression. The main motivation of this thesis is quite general. Thus, the remainder of this section will highlight specific aspects of adaptive format conversion that need further investiga-

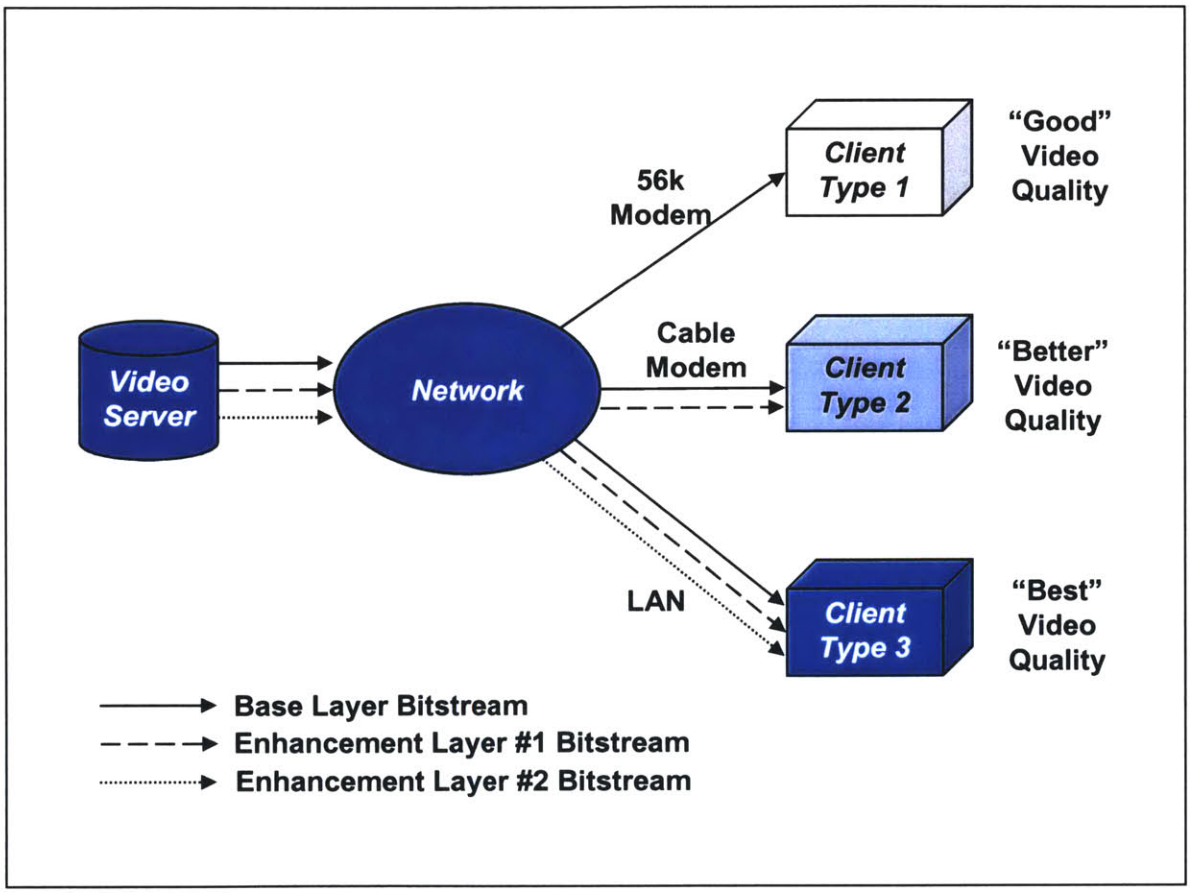

Figure 1.2: An Example of Scalable Coding to Achieve Multiple Levels of Video Service. A three layer scalable coding scheme which satisfies the multicast environment shown in Figure 1.1. Clients with a 56k modem would decode only the base layer and receive "good" video quality. The first enhancement layer could be decoded in addition to the base layer by clients with a cable modem to receive "better" video quality. LAN clients could decode the base layer and both enhancement layers to receive the "best" video quality.

tion to better understand its applicability to video scalability. This overview will provide both motivation and direction for the research of this thesis.

1.2.1 Effect of the Base Layer on Adaptive Format Conversion

The concept of adaptive format conversion has not been studied in detail and therefore is not well understood. Sunshine [7, 8] performed a set of experiments investigating a special case of the general adaptive format conversion system: adaptive deinterlacing for a two service level system. In this system, the base layer was interlaced video and the enhancement layer was progressive video of the same spatial resolution. The main conclusion from his thesis was adaptive deinterlacing can provide a significant improvement in decoded video quality with the transmission of a small amount of enhancement data. A major issue in those simulation results was that the base layer was not coded. This issue has a significant impact on the experimental results and introduces the first motive for this thesis:

Motive #1: What is the effect of base layer coding on adaptive format conversion?

The quality of the base layer is important because any enhancement layer in a scalable coding scheme is dependent on the base layer (as well as previous enhancement layers). Compression of the base layer will introduce distortion in the base layer and this distortion will propagate to the dependent enhancement layers. The lack of base layer coding in the simulations of Sunshine significantly affects the simulation results because this allows the deinterlacing methods to utilize "perfect information" from the remaining fields to recover the missing fields. Therefore, these results can be considered empirical upper bounds on the performance for adaptive format conversion. While the establishment of upper bounds is important and suggests the potential of adaptive format conversion, it is not clear how these compression gains will be affected by coding of the base layer. In fact, there may be insignificant or no compression gain when the base layer is coded at certain qualities. Therefore, the coding efficiency of adaptive format conversion should be reexamined with coding of the base layer. Since compression of the base layer is required in any realistic implementation, this issue must be investigated to ascertain the practicality of adaptive format conversion.

1.2.2 Comparison of Adaptive Format Conversion and Residual Coding

Most scalable coding schemes utilize a fixed method of format conversion for the entire video sequence and residual coding is the only type of enhancement data that is transmitted. Parameters representing adaptive format conversion can be conceptualized as a different type of enhancement data. This suggests the second motive for this thesis:

Motive #2: How does adaptive format conversion compare to residual coding?

One of the differences between the two types of enhancement information is the different amounts of bandwidth that are required for their use. One characteristic of adaptive format conversion is that a smaller number of parameters are often needed for coding a region using adaptive format conversion compared to residual coding. This is because the format conversion method is transmitted in adaptive format conversion compared to a group of quantized coefficients in residual coding. This enables adaptive format conversion to provide video scalability at low enhancement bitrates that are often not possible with residual coding, even with the coarsest quantizer.

Adaptive format conversion and residual coding also have different distortion ranges that they can achieve. The different methods used in adaptive format conversion will have a limit in the degradation that they can reduce. Residual coding does not have this limitation and can recover (practically) all of the video detail, albeit this may require use of a very fine quantizer which will result in extremely high enhancement layer bitrates. The different achievable bandwidths and distortions for adaptive format conversion and residual coding should be examined further and suggest experiments to compare a coding scheme using only adaptive format conversion with another scheme using only residual coding (after use of a fixed method of format conversion for the entire sequence).

1.2.3 Use of Both Adaptive Format Conversion and Residual Coding

Note that adaptive format conversion and residual coding can be conceptualized as different types of enhancement information, but one does not need to choose between them

and can incorporate both types of enhancement information in a scalable compression scheme. This leads to the third motive for this thesis:

Motive #3: Is the joint use of adaptive format conversion and residual coding more efficient than residual coding alone?

Since adaptive format conversion can be used before residual coding, it is interesting to determine whether compression gains can be achieved by adding adaptive format conversion to the "standard" residual coder that most scalable coding algorithms use. That is, comparison of a residual coder that has a fixed method of format conversion to a coder that utilizes both adaptive format conversion and residual coding. This would determine whether adaptive format conversion will improve video scalability at other bitrates besides the low enhancement bitrates not achievable using only residual coding. If inclusion of adaptive format conversion to a residual coder improves performance at higher enhancement rates, this suggests that adaptive format conversion should be included in all scalable coding schemes to improve coding efficiency. If inclusion of adaptive format conversion does not significantly improve coding efficiency at all enhancement rates, it would be important to know the bandwidth ranges where it is inefficient.

1.2.4 Application of Adaptive Format Conversion to the Migration Path

As discussed above, adaptive format conversion has some interesting properties to study for research purposes, but it is also important to consider its practical significance. One important application where adaptive format conversion may have significant impact is the migration path for digital television [7, 9] and this provides the fourth motive for this thesis:

Motive #4: How can adaptive format conversion be applied to the migration path for digital television?

The United States digital television standard has recently been established and provides many substantial improvements over its analog counterpart. However, the standard has significant limitations on the transmittable video formats and the need to migrate to higher resolutions in the future has already been recognized. The concept of a

migration path concerns the transition to resolutions beyond the current standard in a backward-compatible manner. The bandwidth for any enhancement layer is expected to be low in the near future which discourages residual coding, but suggests the use of adaptive format conversion for the migration path. It is unclear whether and when enough additional bandwidth will be allocated to enhancement layers to support residual coding, so the use of adaptive format conversion may be very important for video scalability. If more bandwidth were to become available to support residual coding, it is also important to determine the future role of adaptive format conversion in the migration path.

1.2.5 Summary of Motives

To summarize, the fundamental motive and the four secondary motives of this thesis are listed below. In this work, a number of general results on adaptive format conversion will be discussed that can be utilized in many different scenarios and one specific application (the migration path of digital television) will be investigated in detail.

Fundamental Motive:

- ***When and how can adaptive format conversion be used to improve scalable video compression?***

Secondary Motives:

1. ***What is the effect of base layer coding on adaptive format conversion?***
2. ***How does adaptive format conversion compare to residual coding?***
3. ***Is the joint use of adaptive format conversion and residual coding more efficient than residual coding alone?***
4. ***How can adaptive format conversion be applied to the migration path for digital television?***

1.3 Thesis Overview

Chapter 2 will provide an overview of video coding for multicast environments. Section 2.1 will define video processing terminology that will be used throughout this thesis. Topics that will be discussed include resolution formats, lossy video compression and MPEG history and structure. Single layer coding will be discussed in Section 2.2 since many of the same concepts are also applicable to video coding for multiple layers. A review of multicast video coding techniques in Section 2.3 will provide the main background for this thesis. Simulcast and scalable coding can both be used for multicast video coding, but the focus of this thesis will be on scalable coding since it usually provides higher coding efficiency (with the tradeoff of higher codec complexity).

The information to be coded in the enhancement layers of a scalable coding scheme is an active research area and is introduced in Chapter 3. Section 3.1 introduces the concept of adaptive format conversion information as another type of enhancement data besides residual coding which is well known and the only type of enhancement data in many scalable video compression schemes. A review of previous research on adaptive format conversion is presented in Section 3.2. This review will both suggest the potential benefit of adaptive format conversion and illustrate the need to further investigate when and how this concept can be used to improve scalable video compression. An implementation of an adaptive format conversion system is described in Section 3.3. Section 3.4 discusses the algorithms used for parameter selection in the implementation.

The implementation described in Chapter 3 is used to perform different simulations and the experimental results of these simulations are presented in Chapter 4. Section 4.1 explicitly formulates the scalable video coding problem to be investigated. The results of an experiment with an uncoded base layer are shown in Section 4.2. This experiment will demonstrate both the potential of adaptive format conversion and the advantages of adapting the block size for adaptive format conversion. The next three sections of Chapter 4 address the first three motives of this thesis. The effect of base layer coding on adaptive format conversion is examined in Section 4.3. Section 4.4 compares adaptive format conversion to residual coding. The use of both types of enhancement data for scalable coding is investigated in Section 4.5.

The last motive of this thesis, the application of adaptive format conversion to the

digital television migration path, is examined in Chapter 5. Section 5.1 reviews the U.S. digital television standard and introduces the future need to migrate to higher resolution formats beyond the current standard in a backward-compatible manner. This problem is referred to as the migration path problem and is discussed in Section 5.2. Scalable coding can be used to satisfy the backward-compatibility constraint of the migration path but an additional constraint is the limited amount of bandwidth available to support the enhancement layer. The limited bandwidth discourages residual coding and suggest that adaptive format conversion may be an alternative solution. The application of adaptive format conversion to the migration path problem is discussed in Section 5.3. This section will discuss both the immediate applicability of adaptive format conversion as well as its future role when (and if) more bandwidth becomes available to also support residual coding. Section 5.4 discusses how adaptive format conversion can also be used as enhancement data in a scalable coding scheme for other environments such as cable and satellite as well as other resolutions besides high-definition television.

Chapter 6 summarizes the results of this thesis and describes some future research directions.

Chapter 2

Video Coding for Multicast Environments

This chapter will begin by defining video processing terminology that will be utilized throughout this thesis. Topics to be discussed include resolution formats, lossy video compression and the history and structure of MPEG video coding. A brief review of single layer coding, also known as nonscalable coding, will be provided since many aspects of single layer coding are also applicable when coding video for multicast environments. Multicast video coding approaches can be divided into two types: simulcast coding and scalable coding. Simulcast coding involves coding each representation independently while scalable coding utilizes a single independently coded base layer with other dependent enhancement layers. Scalable coding requires higher codec complexity, but it usually has a higher coding efficiency since enhancement layers can reuse some of the information from previous layers. Since this thesis will approach the multicast video coding problem from the video coding point of view, the emphasis of this thesis will be on scalable coding and different types of scalability can be achieved such as quality scalability, temporal scalability and spatial scalability.

2.1 Video Processing Terminology

2.1.1 Resolution Formats

The video *format* specifies the spatial and temporal resolution of a video sequence. A spatial resolution of C x D represents C lines of vertical resolution with D pixels of horizontal resolution. The temporal resolution of a sequence is defined relative to its *scan mode*. The *scan mode* for a video format is the order in which the pixels of a video sequence are raster scanned. There are two different scan modes used in video processing: *progressive scanning* and *interlaced scanning* (Figure 2.1).

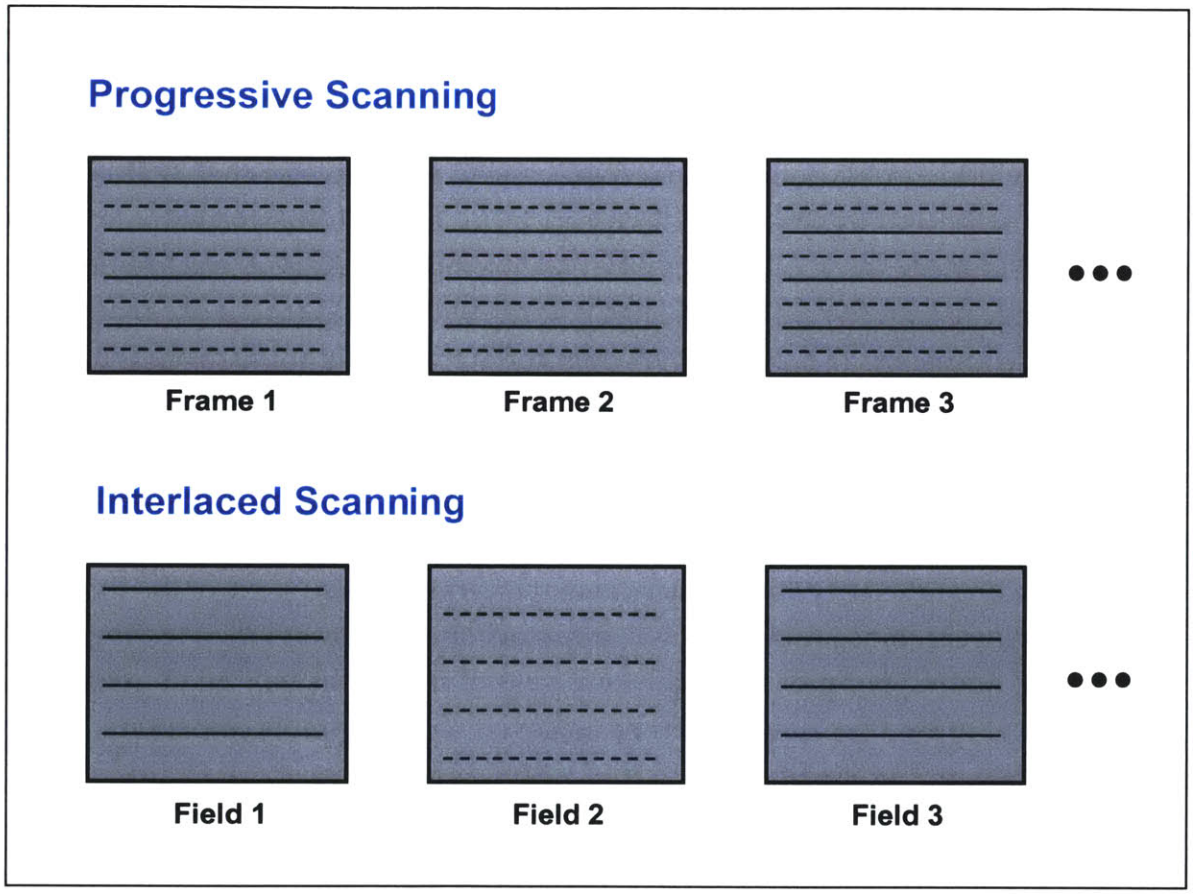

Figure 2.1: Progressive Scanning and Interlaced Scanning. The solid lines represent all the even lines in a frame and the dotted lines represent all the odd lines in a frame. In progressive scanning, every horizontal line in a frame is scanned sequentially from top to bottom. In interlaced scanning, every other horizontal line in a frame is scanned, alternating between the even field (all the even lines) in one frame, the odd field (all the odd lines) in the next frame, the even field in the subsequent frame, etc.

Progressive scanning (PS) is the process of scanning every line, one at a time from left to right and top to bottom in each frame. This is the dominant raster scan used in computer monitors and other high-resolution display devices. Since every pixel in every frame is scanned, this mode will have high spatial resolution and low flicker. The temporal resolution for progressive scanned video is expressed in frames per second (frames/sec).

Interlaced scanning (IS) is an attempt to achieve a tradeoff between spatial-temporal resolution and bandwidth by scanning every other horizontal line in a frame from top to bottom and alternating between the *even lines* and the *odd lines* for each frame.[1] A *field* is defined to be all the lines scanned by interlaced scanning in a single frame. By alternating between the *even field* (the field composed of all the *even lines*) and the *odd field* (the field composed of all the *odd lines*) for each frame, only half of the pixels in every frame are raster scanned in interlaced scanning compared to progressive scanning. This vertical-temporal tradeoff allows slow-moving objects to be perceived at a high spatial resolution. Fast-moving objects are perceived at a high temporal rate, but with half the vertical resolution. The temporal resolution for interlaced scanned video is expressed in fields per second (fields/sec).

Interlaced scanning was first introduced in the analog NTSC standard [10] in an attempt to maximize spatial and temporal resolution while minimizing bandwidth. The vertical-temporal tradeoff of interlaced scanning results in artifacts such as interline flicker, but interlaced scanning artifacts are often overlooked by the human visual system since the eye is not sensitive to the spatial details of fast moving objects. The majority of production and display equipment is currently interlaced due to the analog NTSC standard. Therefore, in spite of the well known artifacts of interlaced scanning, many groups support interlaced video due to economic factors.

2.1.2 Video Compression

All compression algorithms can be classified into two categories: *lossless* and *lossy*. Lossless compression involves reducing the bitrate required to represent the material without

[1] This thesis will use the common convention that the first line is numbered 0 and considered an even line, the second line is numbered 1 and considered an odd line, etc.

any reduction in the information. As one would expect, the bitrate reduction from lossless compression is often limited. Examples of lossless compression include Lempel-Ziv coding [11, 12] and Run-Length Encoding. Lossy compression can include some information reduction which enables a much higher reduction in the bitrate representation than lossless compression. In addition, the reduction in decoded quality can often be undetectable or minor when insignificant details are removed by the lossy compression.

A video sequence usually occupies a vast amount of bandwidth, making it impractical for most applications to transmit and/or store sequences without any compression. Lossless compression can be utilized, however the resulting bitstreams are often still much larger than the available bandwidth or storage capacity. In addition, certain visual characteristics can be exploited to minimize the perceptual loss of quality with lossy compression, therefore, almost all video transmission and storage applications utilize lossy compression. The use of any lossy compression requires that a measure of the decoded video quality be established to ascertain the tradeoff between quality and bandwidth. This thesis will use the Peak Signal-to-Noise Ratio (PSNR) as a quantitative measure of the decoded video quality where PSNR is defined as

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{2.1}$$

and expressed in decibels (dB). The Mean Square Error (MSE) is defined to be the average squared difference between the luminance components of the original and decoded video. Note that the quantitative measure of PSNR may not exactly correspond to the perceived quality of the human visual system. The lack of a "standard" measure is a common problem in many image and video processing studies. Therefore, one may consider these experiments (as well as any experimental results which use a quantitative measure for the human visual system) as an approximation to the results that would be perceived by human viewers. Nevertheless, the use of PSNR as a quantitative measure of quality is a common practice in the video processing field and has been found to be a sufficient measure for many studies. In addition, similar experiments can easily be performed using a different quantitative criteria if so desired.

2.1.3 MPEG History

Considerable research has been performed on the efficient digital representation of video signals over the past 25 years. This technology has resulted in products for a wide range of applications such as digital video discs (DVDs), digital television/high definition television and video over the Internet. The need for international compression standards arose to meet the increased commercial interest in video communications. A very popular set of standards was developed by the Moving Pictures Experts Group (MPEG). MPEG is the working group ISO/IEC JTC 1/SC 29/WG 11 within the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). This group has been developing coding standards for digital audio and digital video since 1988. Their efforts have led to three multimedia standards: MPEG-1 (formally referred to as ISO/IEC 11172), MPEG-2 (formally referred to as ISO/IEC 13818) and MPEG-4 (formally referred to as ISO/IEC 14496).[2]

While there are many differences between the three standards, the most distinguishing characteristics between them involve the intended bitrates and applications. MPEG-1 [13, 14] is intended for intermediate bitrate applications (on the order of 1.5 Mbits/sec) such as storage on a CD-ROM. MPEG-2 [2, 3] is intended for higher bitrate applications (10 Mbits/sec or more) such as television broadcasting (cable, satellite and terrestrial). MPEG-2 also provides syntax for both interlaced video signals and scalable video signals. These capabilities were nonexistent in MPEG-1 which was limited to non-scalable progressive (noninterlaced) video pictures. MPEG-4 [4, 15] was originally intended for very low bitrate applications (about 64 kbits/sec) such as video telephony. Its focus has changed over time to include much more than very high compression. One new aspect of MPEG-4 is the capability to compress arbitrarily shaped video objects. This was not possible in the earlier two standards which were limited to rectangular shaped video. The compression of video objects (instead of pictures) introduces many new functionalities such as content-based manipulation, scalability and editing.

[2] A fourth standard (MPEG-7) is currently being developed by MPEG and is intended for video indexing and browsing.

2.1.4 MPEG Structure Nomenclature

The three existing MPEG standards use the same basic structure to define the different layers of a video sequence. This thesis will utilize the MPEG structure nomenclature which is well known in the video processing research field. The MPEG standard defines many layers and the definition of these layers are useful for ease of discussion in the remainder of this thesis. The outermost layer is the *video sequence* layer which consists of a set of individual *pictures* occurring at fixed time increments. Since a sequence may be composed of many different scenes, subsets of the video sequence are often grouped together for compression. A subset of pictures is referred to as a *group of pictures* (GOP). In a *closed* GOP, the pictures are predicted only from other pictures in the GOP. In an *open* GOP, pictures may also be predicted from pictures not in the GOP. The *closed* definition is commonly assumed with the GOP terminology. This thesis will follow this convention, therefore every GOP will implicitly be closed.

Each picture in the sequence is the same size and is composed of three *components* to represent the color. There are many ways to represent color with three components. This thesis will discuss two approaches: the RGB component system and the YUV component system. The RGB system is a common system for representing colored pictures and consists of additive Red, Green and Blue components. This component system is typically used by video capture and display devices since the additive color components are simple to capture and generate with the appropriate color specific filters. The YUV system is a different component system which separates the *luminance* and *chrominance* components of a picture. The Y component represents the *luminance*, the brightness of a picture. The U and V components represent the *chrominance*, the hue and saturation of a picture. The RGB and YUV color spaces are related by the following matrix equation:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.2990 & 0.5870 & 0.1140 \\ -0.1687 & 0.3313 & 0.5000 \\ 0.5000 & 0.4187 & 0.0813 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.2}$$

The main advantage of the YUV component system is that the luminance and chrominance are separated and can be processed independently. The human visual system is sensitive to high frequency luminance components, but insensitive to high frequency chrominance components. This characteristic can be exploited by subsampling

the chrominance component pictures and is the reasoning behind the use of the YUV component system for transmission and storage to reduce the bandwidth required to represent a video sequence.

Each component of a picture is a two-dimensional (2-D) array of *pixels* that may not have the same resolution as the composite picture. MPEG defines three different chrominance formats: 4:4:4, 4:2:2 and 4:2:0. The 4:4:4 format does not involve any chrominance subsampling, so each component is the same resolution as the composite picture. In the 4:2:2 and 4:2:0 formats, the luminance component has the same dimensions as the composite picture, but the chrominance components are subsampled. The chrominance components in the 4:2:2 format have the same number of rows as the luminance component, but the columns are subsampled by two. The chrominance components in the 4:2:0 format have both the rows and columns subsampled by two compared to the luminance component. Thus, video in the 4:2:2 and 4:2:0 formats have 67% and 50% of the pixels as video in the 4:4:4 format. Figure 2.2 illustrates an example of how chrominance subsampling can be used to save bandwidth for video transmission.

The next two layers of a video sequence are the *slice* and *macroblock* layers. These layers are necessary since a picture usually does not have stationary characteristics across the whole scene. This suggests subdividing a picture to exploit local stationarity and many video compression algorithms including the MPEG compression standard utilize *macroblocks*. Macroblocks are the standard building blocks of MPEG pictures and are defined to be a 16x16 array from the luminance component along with the corresponding arrays from the chrominance components. Macroblocks are nonoverlapping and numbered in raster scan order. A *slice* is defined to be a contiguous sequence of macroblocks in raster scan order. A *block* is defined as a single 8 x 8 array in any component image. Therefore, a macroblock from a picture with the 4:2:0 chrominance format consists of four 8 x 8 blocks from the luminance component and one 8 x 8 block from each of the chrominance components (Figure 2.3).

The definition of the many layers (video sequence, group of pictures, picture, macroblock, block) in a MPEG system allows a great deal of flexibility for defining and changing coding parameters.

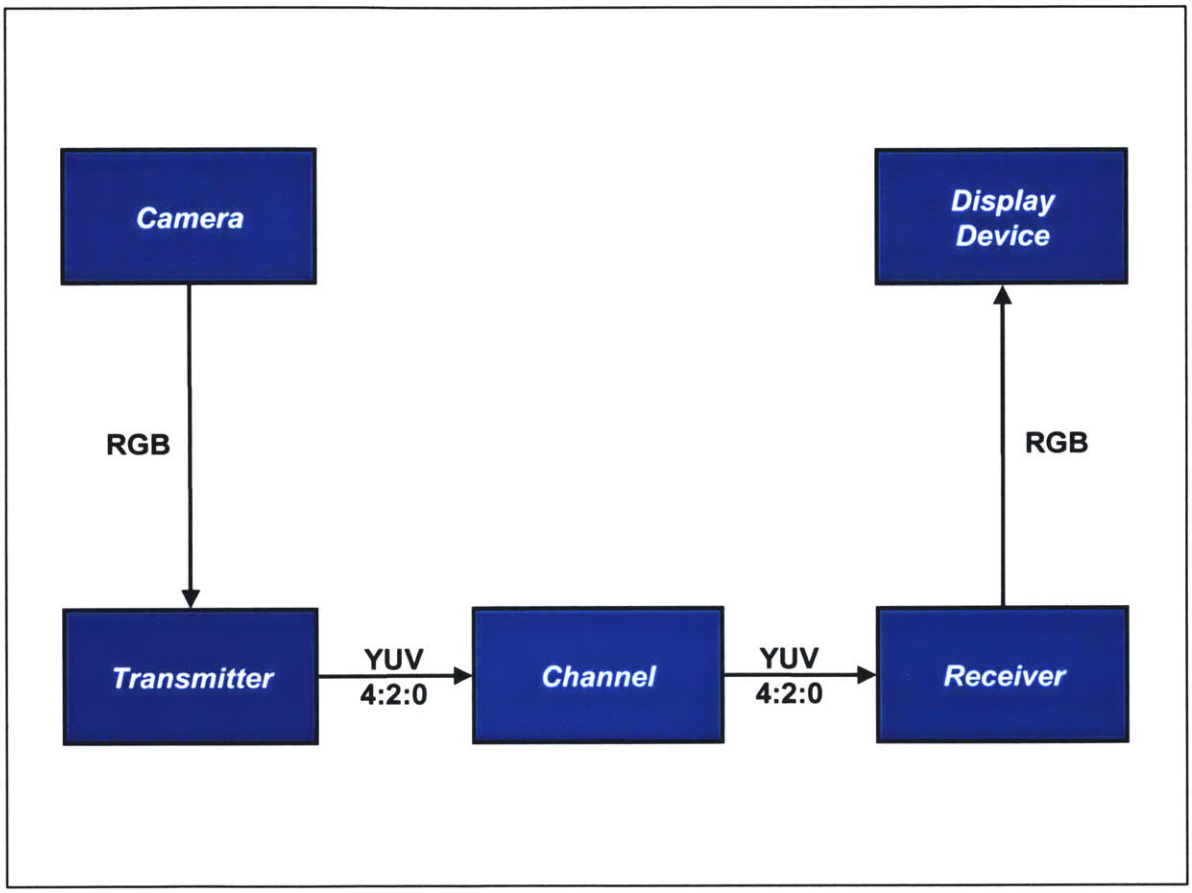

Figure 2.2: Chrominance Subsampling to Save Transmission Bandwidth. A camera captures video in the RGB format and sends it to the transmitter which converts it to the 4:2:0 YUV format. The 4:2:0 YUV video is transmitted across a channel and the receiver converts it back to the RGB format after reception. The RGB video is then sent to a display device. Transmission of 4:2:0 YUV video requires half the bandwidth compared to transmission of the original RGB video.

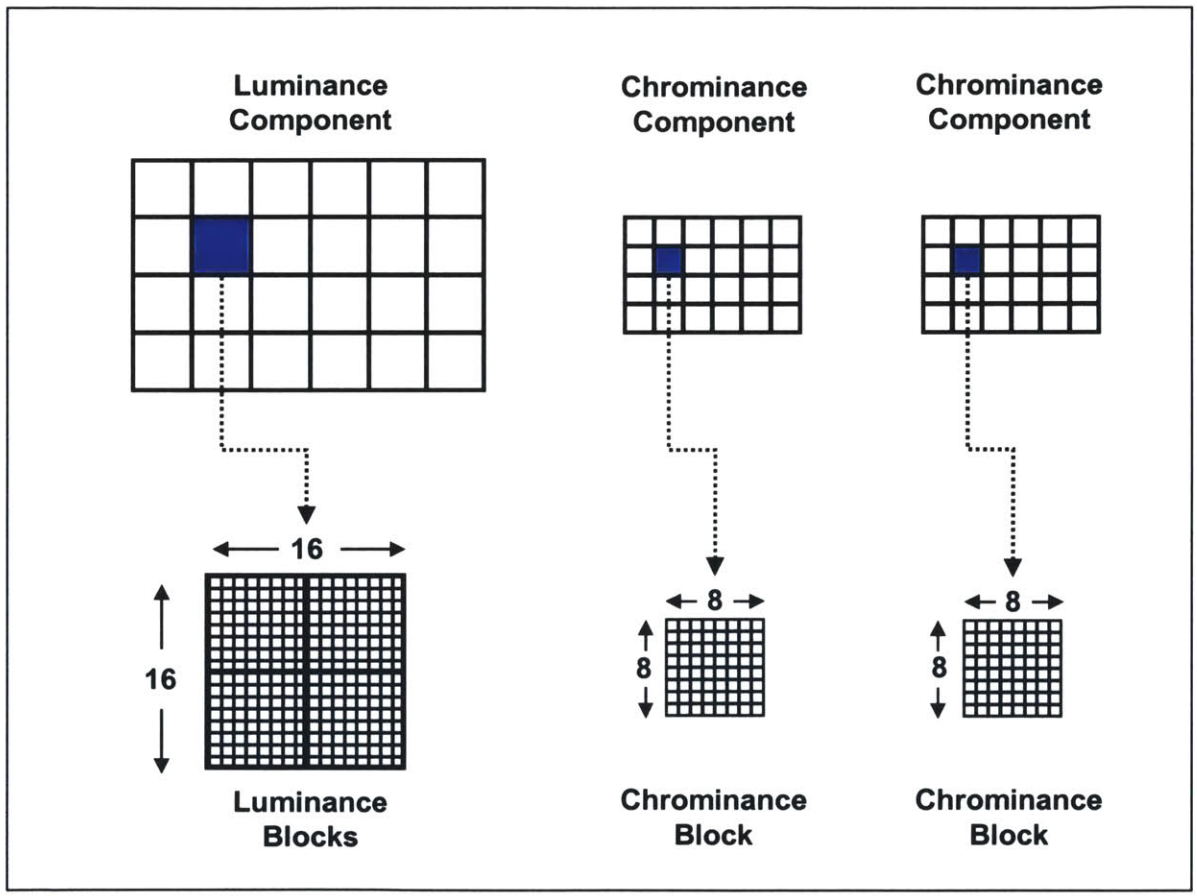

Figure 2.3: A MPEG Macroblock. The macroblock of a picture with the 4:2:0 chrominance format consists of four 8 x 8 blocks from the luminance component and one 8 x 8 block from each of the chrominance components.

2.2 Single Layer Coding

The focus of this thesis is on multicast video coding, the coding of a single video sequence at more than one resolution format and/or quality. Before reviewing multicast coding techniques, it is instructive to briefly review some basic concepts in single layer coding since many of the concepts are also applicable when coding for multiple layers. To illustrate the relation between single layer coding and multicast coding, consider the development of the multicast coding techniques in the MPEG standards. The first MPEG standard, MPEG-1, specifies syntax only for nonscalable (single layer) coding. The scalable tools in MPEG-2 and MPEG-4 are essentially extensions of the basic single layer coding principles established in MPEG-1. MPEG single layer coding exploits two forms of redundancy inherent in any video sequence: *spatial redundancy* and *temporal redundancy*.

Many methods have been proposed to efficiently compress images and video sequences by exploiting the inherent spatial redundancy. The most popular has been the use of the discrete cosine transform (DCT) on nonoverlapping macroblocks. The energy compaction properties of the DCT are widely known [16] and an additional benefit is that many fast, low-cost implementations have been developed, making DCT-based compression algorithms very popular for hardware implementations.

The DCT is applied on each block transforming the intensities to frequency coefficients. These coefficients are then *quantized* and transmitted instead of transmitting the original image intensities. The result of *quantization* is that some coefficients are represented with less precision and therefore occupy less bandwidth. Each coefficient is quantized by dividing it by a nonzero positive integer called a *quantizer* and the quotient is rounded to the nearest integer. The rounded quotient, known as the quantized coefficient, is transmitted and can be used (with the value of the quantizer) to reconstruct an approximation to the original coefficient value. In addition to the energy compaction property of the DCT, the human visual system is more sensitive to lower frequencies, so low frequency components are quantized more finely than high frequency components. This is done by using frequency dependent quantizers with larger quantization values for higher frequencies so they are represented more coarsely than lower frequencies. Pictures where every macroblock is encoded in this manner are referred to as *intra-coded* pictures or *I-pictures*.

Temporal redundancy of a video sequence can be exploited by applying motion compensation. This is performed at the macroblock level to predict a macroblock in the current picture from adjacent pictures. The MPEG compression standard allows two types of motion compensation to occur: predictive-coded pictures (*P-pictures*) and bidirectionally predictive-coded pictures (*B-pictures*). *P-pictures* are coded with respect to a previous *I-picture* or *P-picture* and therefore may involve the transmission of up to one motion vector per macroblock. Note that the use of predictive coding is optional and not required for each macroblock. Thus, an encoder may choose to encode certain macroblocks in a *P-picture* without any prediction. *B-pictures* are coded with respect to one previous and one future *I-picture* or *P-picture* and therefore may involve transmission of up to two motion vectors per macroblock. Similar to *P-pictures*, the use of bidirectional predictive coding is optional, therefore an encoder may choose to encode macroblocks in a *B-picture* without any prediction, using predictive coding or using bidirectional predictive coding. Also note that motion compensation is never performed using another *B-picture* as the reference picture. Any pictures that use motion compensation are referred to as *inter-coded* pictures.

Motion compensation in inter-coded pictures is applied to obtain an estimate for the macroblock to be coded. The difference between the desired macroblock and its estimate is called the *residual*. The residual is encoded by applying the DCT operation followed by quantization and transmission of the quantized coefficients in a manner similar to macroblocks in intra-coded pictures. The decoded residual can then be used at the decoder along with the appropriate motion vectors to reconstruct the macroblock. The use of motion compensation is optional for every macroblock in an inter-coded picture. Macroblocks that do not use motion compensation are coded in the same manner as those in intra-code pictures. Figure 2.4 shows a high-level diagram of a MPEG codec.

The MPEG parameters M and N define the picture structure. The parameter M defines the number of pictures between I-pictures and P-pictures. The parameter N defines the number of pictures in a GOP. The use of closed GOPs means that N can also be interpreted as the number of pictures between consecutive I-pictures. Figure 2.5 shows the MPEG picture structure with $M = 3$ and $N = 6$. The arrows in the figure indicate the reference pictures that are used for motion compensation in the inter-coded pictures.

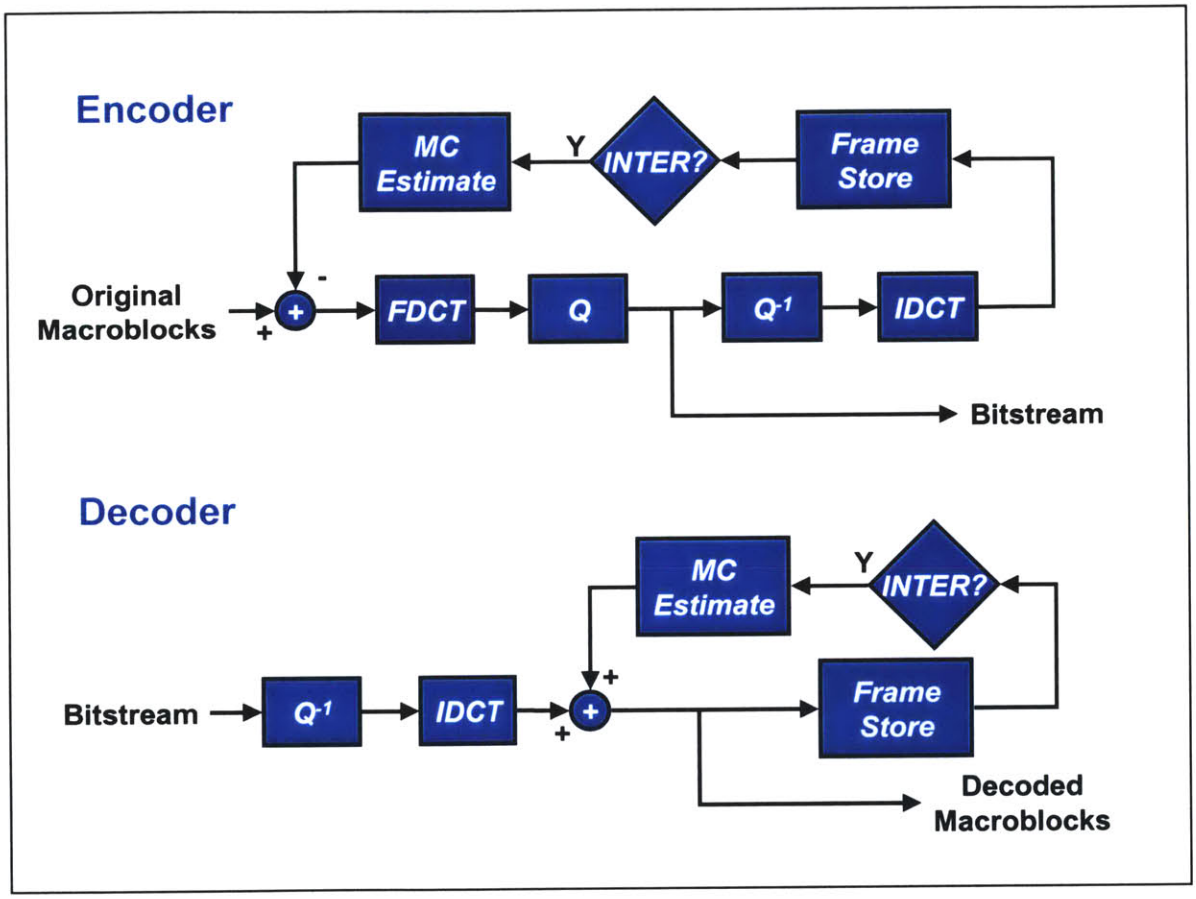

Figure 2.4: High-Level Diagram of MPEG Codec. The input to the encoder are the macroblocks of the original video sequence. A forward DCT ($FDCT$) is applied to the original macroblock or the residual depending on whether the macroblock is to be intra-coded or inter-coded, respectively. The DCT coefficients are then quantized (Q) and transmitted as part of the bitstream. The encoder mimics the decoder to provide its own version of the reconstructed macroblocks for inter-coded macroblocks. The decoder dequantizes (Q^{-1}) the quantized coefficients and then applies an inverse DCT ($IDCT$) to obtain the reconstructed macroblock. This macroblock is then stored for possible future use in motion compensation ($INTER?$). The decoder simply inverts the operations of the encoder.

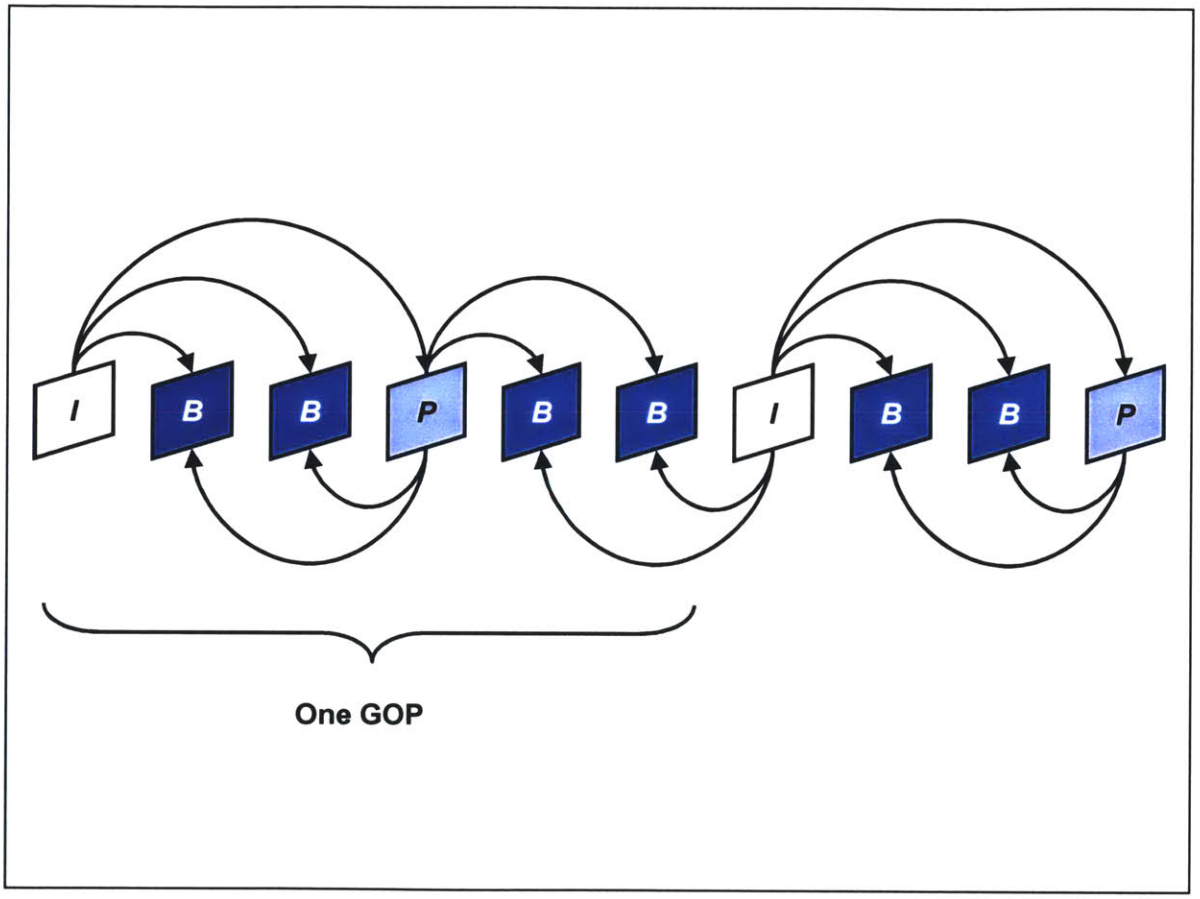

Figure 2.5: MPEG Picture Structure With $M = 3$ and $N = 6$. The arrows indicate the pictures that predictive and bidirectionally predictive motion compensation can reference.

2.3 Review of Multicast Video Coding Techniques

Many video applications are designed for single layer coding and decoding, the transmission and reception of video at a single resolution format. However, it may be desirable to have the capability to receive a video sequence at different formats and/or qualities commensurate with the available bandwidth, processing power and/or memory resources of the receiver. [17, 18, 19] Coding techniques to transmit and receive video at different formats and/or qualities are classified into two categories: *scalable coding* and *simulcast coding* (Figure 2.6). *Scalable coding* is the process of encoding video into an independent *base layer* and one or more dependent layers, commonly termed *enhancement layers*. This allows some decoders to decode the base layer to receive basic video and other decoders to decode enhancement layers in addition to the base layer to achieve higher temporal resolution, spatial resolution and/or video quality. [20, 21, 22] Note that additional receiver complexity beyond single layer decoding is required for scalable decoding. *Simulcast coding* involves coding each representation independently and is usually less efficient than scalable coding [23] since similar information in another bitstream is not exploited. The bitstreams are decoded independently, therefore, unlike scalable coding, additional decoder complexity beyond single layer capability is not required for simulcast decoding.

2.3.1 Simulcast Coding

One method to transmit video at multiple resolutions and/or qualities is *simulcast coding*. Figure 2.7 shows simulcast coding with two bitstreams. This involves coding each representation independently and multiplexing the coded bitstreams together for transmission. The demultiplexed bitstreams are decoded independently at the receiver exactly as in the single layer coding case. Therefore, no additional decoder complexity beyond single layer capabilities is required to decode any video sequence when utilizing simulcast coding. This may be important for some commercial applications since additional decoder complexity often increases the cost of receivers.

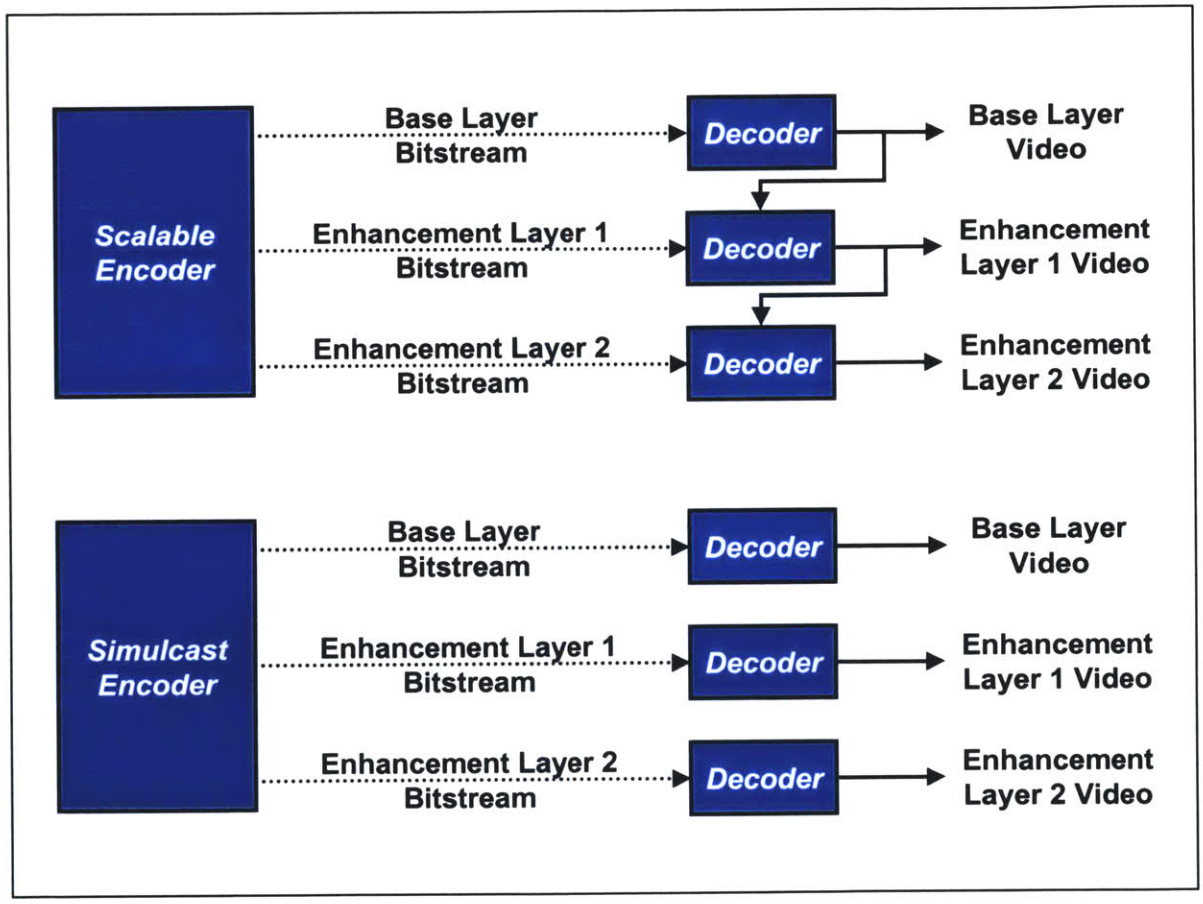

Figure 2.6: Scalable Coding and Simulcast Coding. Examples of scalable coding and simulcast coding for three layers or levels of service. Scalable coding has one independently coded base layer with two enhancement layers that are dependent on the base layer. (Note that decoding of Enhancement Layer 2 is also dependent on Enhancement Layer 1.) All of the layers in simulcast coding are independently coded.

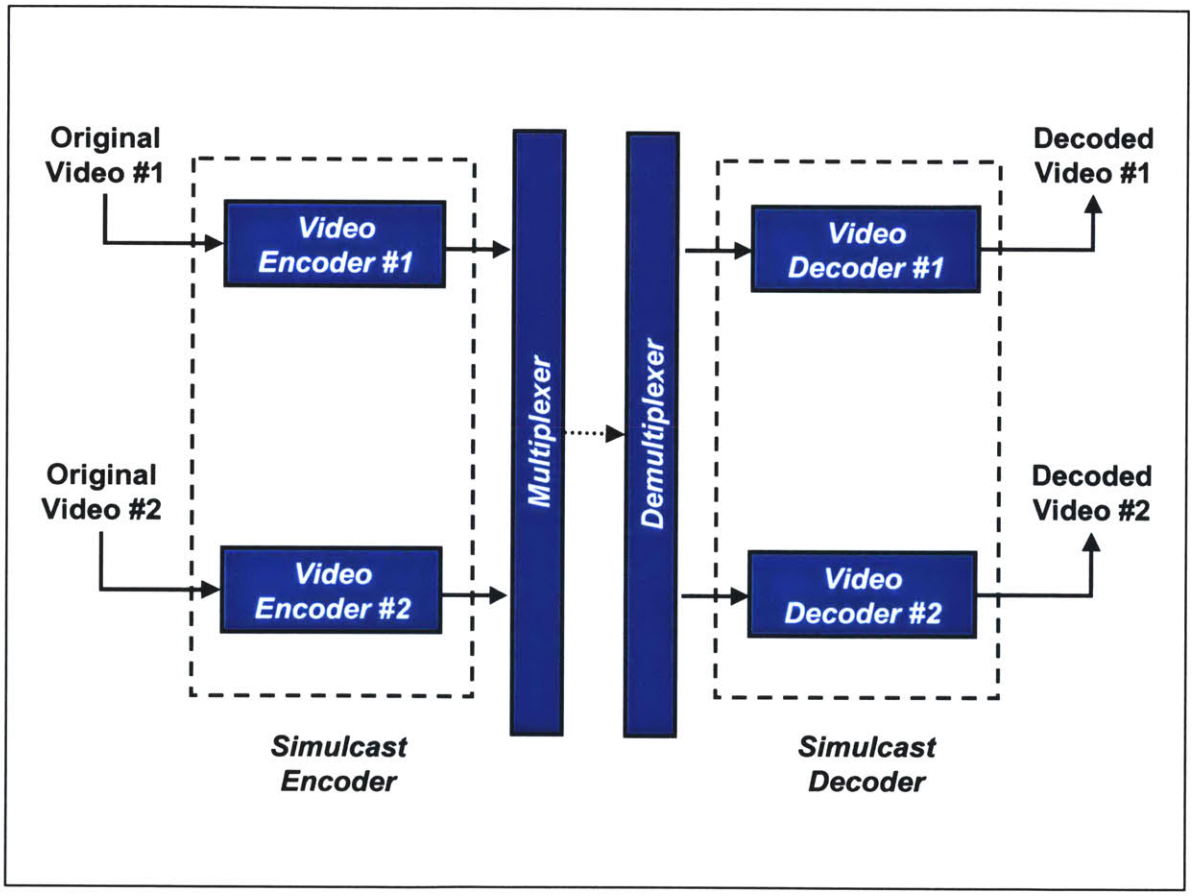

Figure 2.7: Simulcast Coding For Two Bitstreams. Both video sequences are coded and decoded independently as in the single layer coding case.

2.3.2 Scalable Coding

The second method to transmit video at multiple resolutions and/or qualities is *scalable coding*. In general, scalable coding can involve multiple layers or levels of service, however, this thesis will focus on the two layer service system. Results obtained with two service levels will be applicable to scalable systems with more levels. The use of two layers allows the unambiguous definition of the *base layer* as the independently coded layer and the *enhancement layer* as the dependently coded layer. Figure 2.8 shows scalable coding with two bitstreams. The encoding and decoding of the base layer video operates in the same manner as single layer coding. Enhancement layers are constructed with a midprocessor which effectively mimics a standard decoder and then uses the decoded information for prediction in the enhancement layer.

Scalable coding is usually, but not always, more efficient than simulcast coding [23] at the expense of additional complexity. The additional complexity required by scalable coding during encoding and decoding is an important issue since it would increase the cost of both transmitters and receivers. This issue may be important for many applications. This thesis will adopt the video coding perspective and ignore the cost of codec complexity in its analysis.

There are three types of scalable coding: *quality scalability*, *temporal scalability* and *spatial scalability* that increase the picture quality, temporal resolution and spatial resolution, respectively, of the decoded video when an enhancement layer is utilized. The first MPEG standard to define syntax for scalable video was MPEG-2. The main commercial applications that MPEG-2 targeted were digital video disks and digital television, applications where the additional functionality of scalability is often not utilized. Thus, there has been limited commercial interest in MPEG-2 scalable coding in the past. However, new applications such as streaming video could greatly benefit from scalability and have sparked interest in scalable coding. In addition to scalability at the frame level, the recently completed multimedia standard MPEG-4 (Version 1) also defines syntax for scalability of arbitrary shaped objects. This is an interesting new area, however, for simplicity this thesis will only deal with frame based scalability. Each of the scalable coding types will be briefly reviewed in the following text.

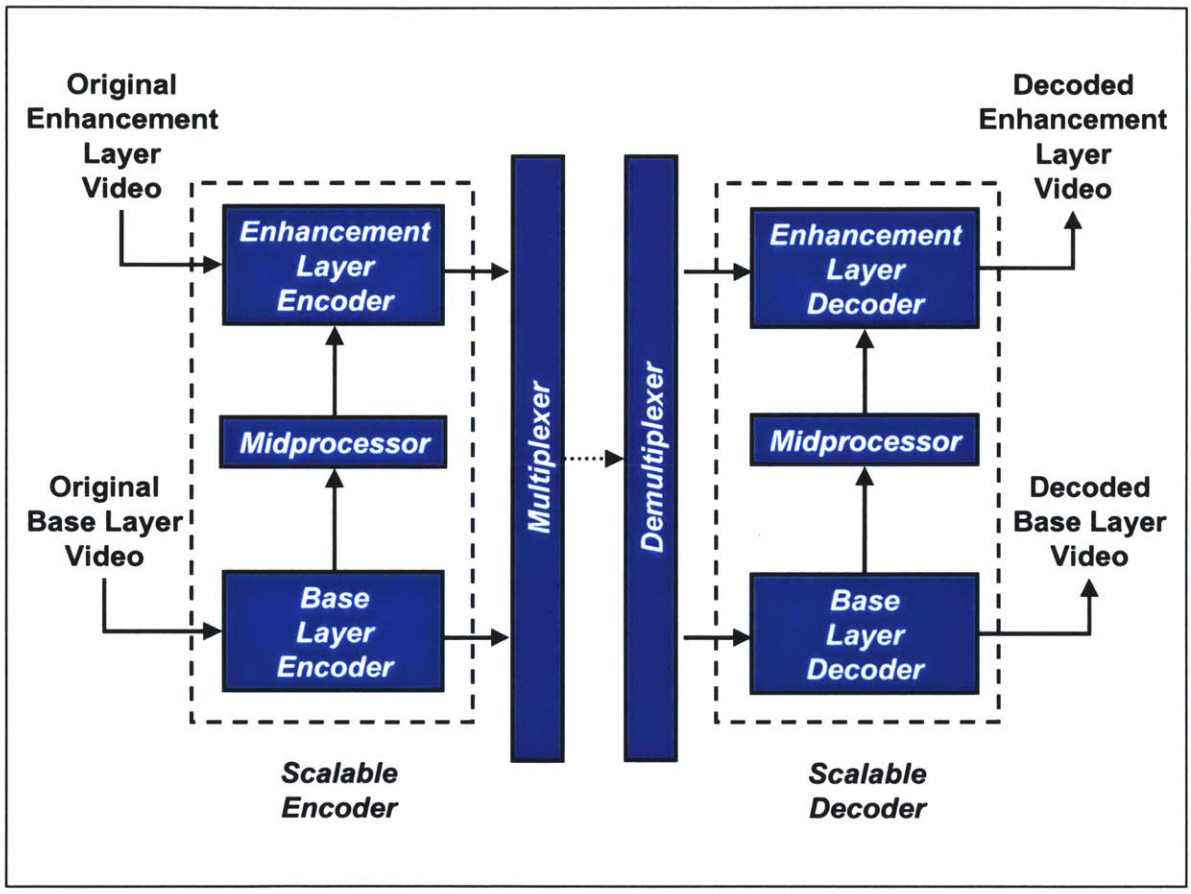

Figure 2.8: Scalable Coding For Two Bitstreams. The base layer is coded and decoded independently as in the single layer coding case. The coding and decoding of the enhancement layer is dependent on the base layer since midprocessors at both the encoder and decoder utilize the base layer to assist coding of the enhancement layer.

2.3.2.1 Quality Scalability

Quality scalability permits an increase in the picture quality by using enhancement layers in addition to the base layer. Figure 2.9 shows quality scalability with two layers. Basic video is obtained by decoding only the independent base layer, which would be done in the same manner as in the non-scalable, single layer case. Decoding of the dependent enhancement layer gives higher quality video with the same spatial and temporal resolution. Quality scalability was implemented in MPEG-2 with the SNR Scalability profile. Another form of quality scalability called Fine Granular Scalability (FGS) is currently being evaluated for inclusion in MPEG-4. [24]

2.3.2.2 Temporal Scalability

Temporal scalability permits an increase in the temporal resolution by using enhancement layers in addition to the base layer. Figure 2.10 shows temporal scalable coding with two layers. Basic video is obtained by decoding only the independent base layer, which would be done in the same manner as in the non-scalable, single layer case. In this example, use of the dependent enhancement layer gives video with three times the temporal resolution of the basic video. The same spatial resolution is obtained whether or not the enhancement layer is decoded. A frame in the enhancement layer can utilize motion compensated prediction from the previous or next frame in the display order belonging to the base layer as well as the most recently decoded frame in the same layer.

2.3.2.3 Spatial Scalability

Spatial scalability permits an increase in the spatial resolution by using enhancement layers in addition to the base layer. Figure 2.11 shows spatial scalable coding with two layers. Basic video is obtained by decoding only the independent base layer, which would be done in the same manner as in the non-scalable, single layer case. In this example, use of the dependent enhancement layer gives video with twice the spatial resolution of the basic video. The same temporal resolution is obtained whether or not the enhancement layer is decoded. A frame in the enhancement layer can utilize motion compensated prediction from the temporally coincident frame in the base layer as well as the most recently

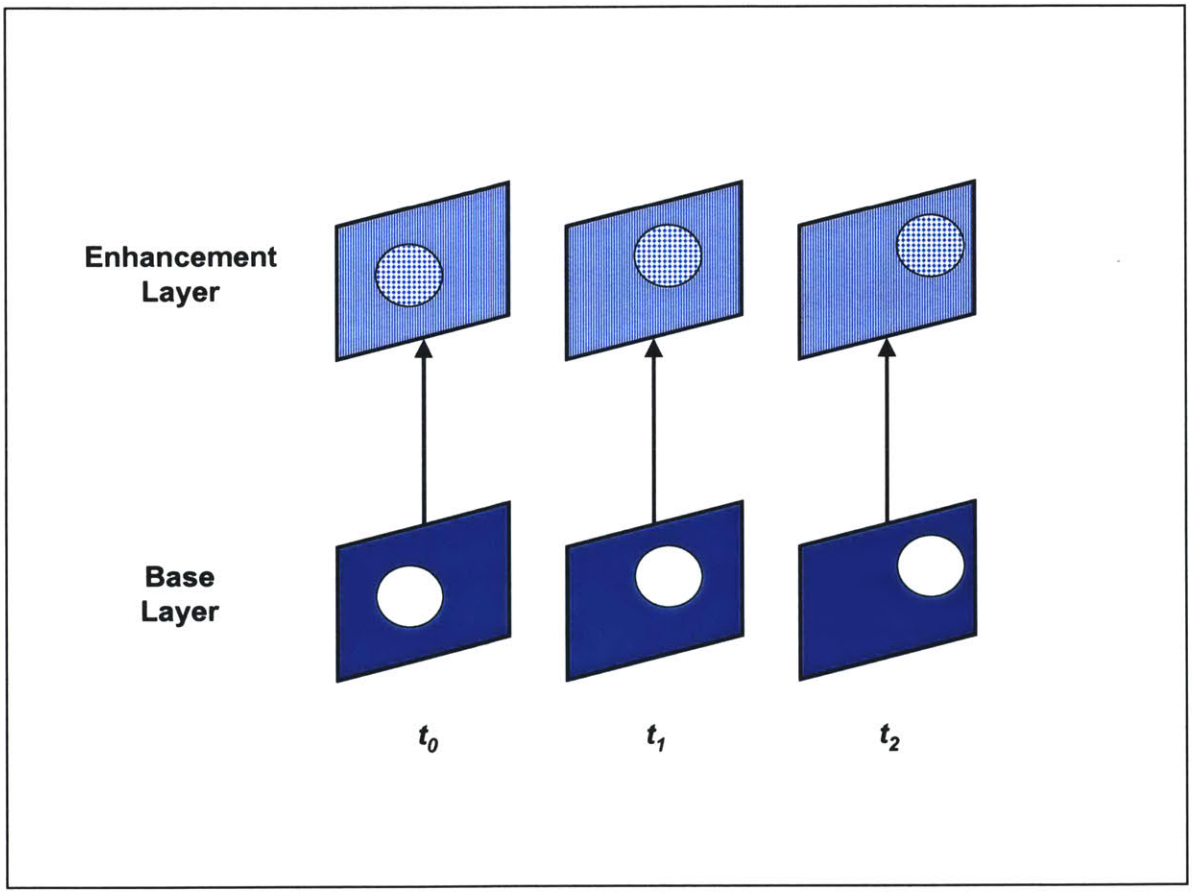

Figure 2.9: Quality Scalability For Two Bitstreams. Decoding of the enhancement layer results in video with the same spatial and temporal resolution as the base layer, but with higher quality, i.e. detail.

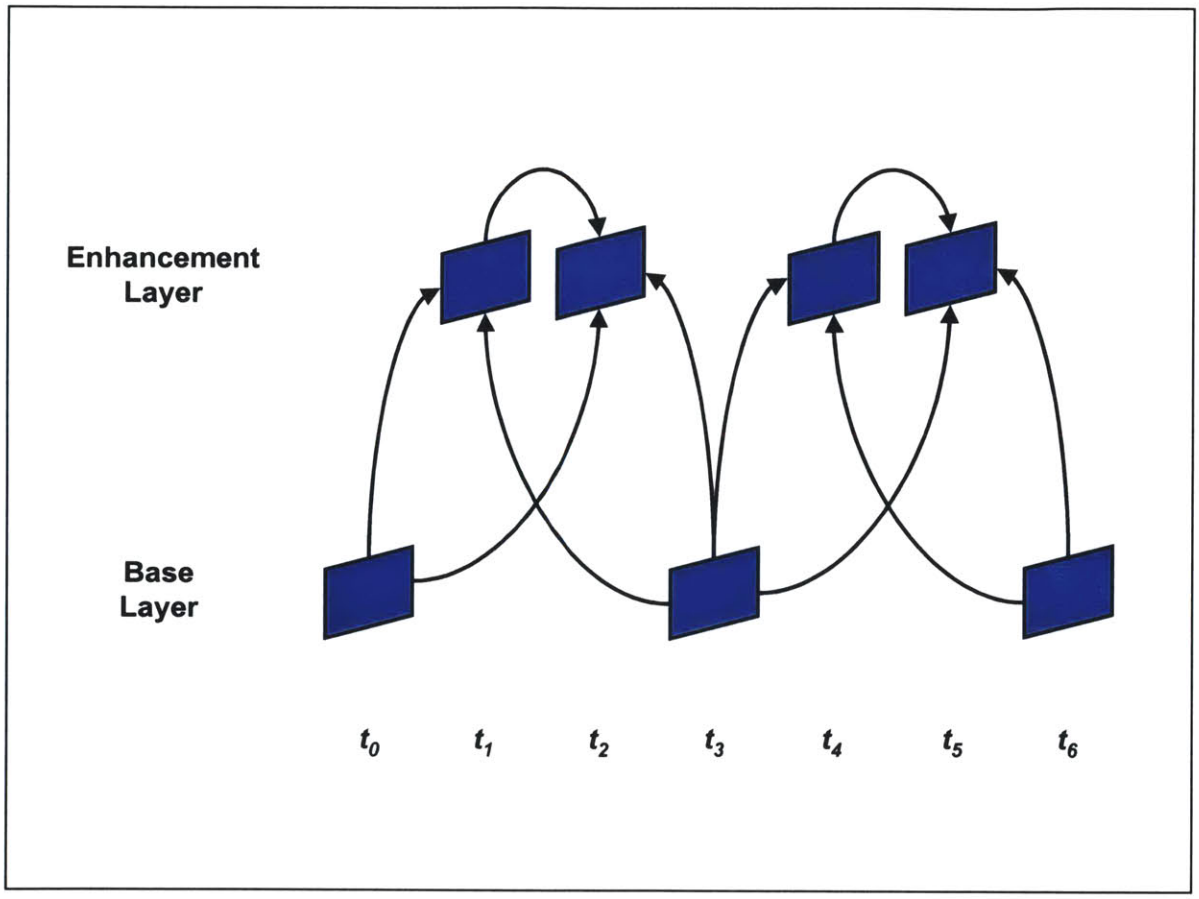

Figure 2.10: Temporal Scalability For Two Bitstreams. In this example, decoding of the enhancement layer results in video with the same spatial resolution as the base layer and three times the temporal resolution.

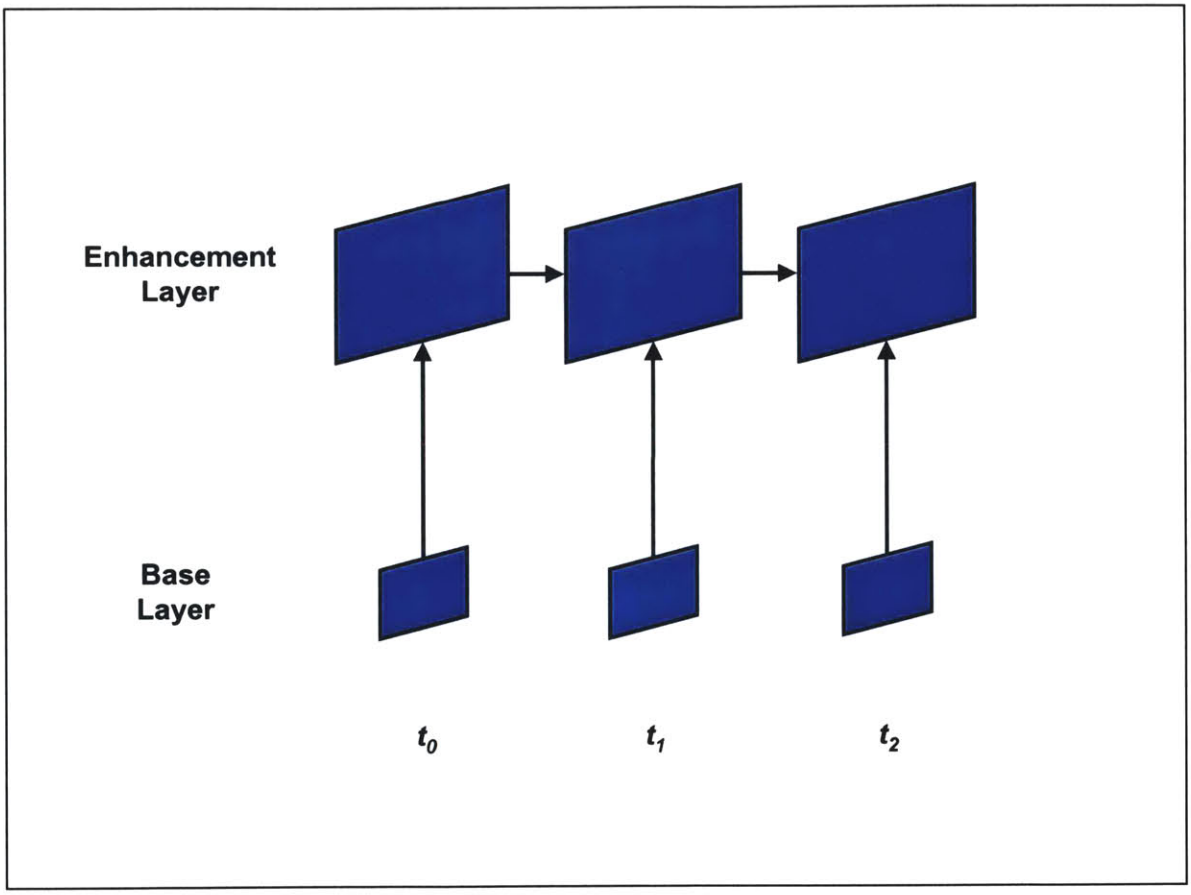

Figure 2.11: Spatial Scalability For Two Bitstreams. In this example, decoding of the enhancement layer results in video with the same temporal resolution as the base layer and twice the spatial resolution.

decoded frame in the same layer.

A special case of spatial scalability is interlaced-progressive spatial scalability and this specific scenario will be discussed in more detail later in this thesis. Figure 2.12 shows an example of this type of scalability with two layers where the base layer is interlaced scanned video and the enhancement layer is progressive scanned video with the same spatial resolution. Deinterlacing will be necessary to convert the interlaced base layer to the progressive enhancement layer.

2.4 Summary

This chapter began by defining video processing terminology that will be used throughout this thesis. After discussing resolution formats and lossy video compression, a review of the history and structure of MPEG video coding was provided. The MPEG video system is well known in the video compression field and the definition of the many layers in the system provide a great deal of flexibility for defining and changing coding parameters. Coding techniques for single layer video coding were then discussed followed by a review of multicast video coding. Multicast video coding can be classified into two categories: simulcast coding and scalable coding. Scalable coding usually has a higher coding efficiency than simulcast coding since enhancement layers can exploit information in the base layer or previously coded enhancement layer(s). The higher coding efficiency is achieved with a tradeoff of increased codec complexity. Following the video coding point of view, this thesis will focus on the use of scalable coding to provide service to multicast video environments. Each enhancement layer of a scalable coded bitstream can increase the quality, temporal resolution or spatial resolution of the decoded video. A special case of spatial scalability is interlaced-progressive spatial scalability and this specific scenario will be examined in more detail later in this thesis.

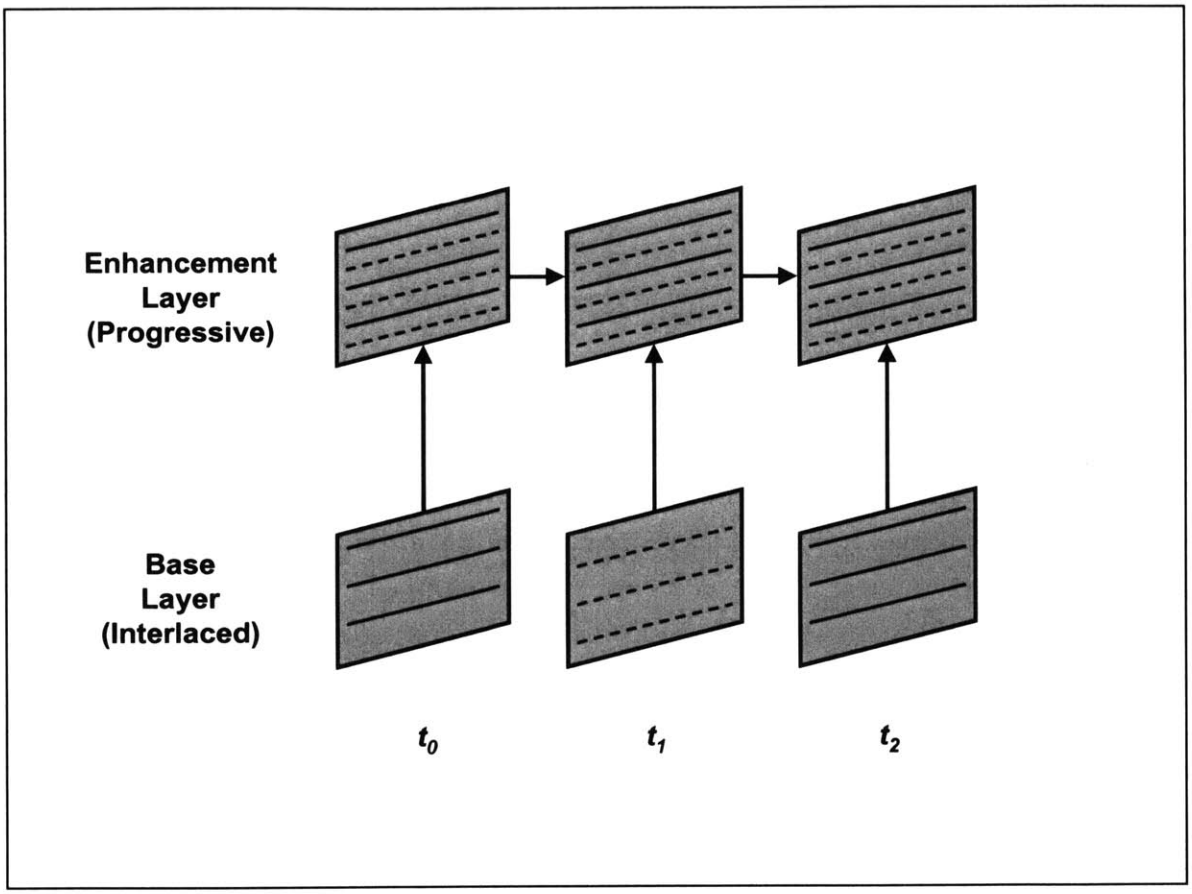

Figure 2.12: Interlaced-Progressive Spatial Scalability For Two Bitstreams. Decoding of the enhancement layer results in progressive scanned video with the same spatial resolution as the interlaced scanned video of the base layer.

Chapter 3

Adaptive Format Conversion

The previous chapter introduced the multicast video coding problem and showed how a scalable coding framework can be used to efficiently provide service to this environment. The concept of scalable coding is well-known in the video compression field, but there is a continued interest in increasing the coding efficiency of enhancement layer bitstreams. This chapter will begin by examining this problem and it will be shown that there are two basic types of information that can be coded in the enhancement layer of a scalable coding scheme. A well-known type of enhancement information is residual coding, which is used in scalable coding schemes such as the spatial scalability profiles in the MPEG-2 and MPEG-4 multimedia standards. There is another type of information that can be used instead of or in addition to residual coding. Recent research has shown that adaptive format conversion information may be able to improve video scalability, but it has not been studied in detail. The main motivation of this thesis is to evaluate adaptive format conversion and determine when and how it can improve video scalability. A review of the previous work in this research area will show limitations of previous implementations and demonstrate the need to develop a new implementation to evaluate the potential of adaptive format conversion. The remainder of this chapter will provide details on the implementation developed for this thesis.

3.1 Information to Encode in an Enhancement Layer

The information to encode in an enhancement layer is a very important issue in any scalable coding scheme. For ease of discussion, this thesis will always assume a scalable framework with only two layers. In general, scalable coding schemes can have many layers resulting in an exponential amount of dependency since each enhancement layer is dependent on the previously coded layer. The two layer case is the simplest example of

scalable coding and consists of one independently coded base layer and one dependently coded enhancement layer. This reduction of complexity is not as significant as it might seem since many of the results obtained with a two layer system are applicable to scalable systems with more than two layers. This can be seen by noting the recursive structure of the scalable coding framework and considering the previously coded layer to act as the "base layer" for an enhancement layer.

Figure 3.1 illustrates the transmitter of a scalable coding system with two layers. The input video (which is in the enhancement layer format) is first converted to the base layer format and sent to the base layer encoder to generate the base layer bitstream. After the base layer is independently coded, the transmitter can mimic a base layer decoder and decode the base layer bitstream. The decoded base layer can then be used along with the original input video to create two types of enhancement data. The first type of enhancement data is information about the signal processing that converts the base layer video into the enhancement layer video format. Note that no format conversion is necessary for quality scalability since both layers have the same resolution. Therefore, it is not possible to transmit this type of enhancement information in quality scalability. Many scalable techniques such as the spatial scalability profiles in the MPEG-2 and MPEG-4 standards convert the base layer in a fixed manner and choose not to utilize this type of enhancement information, but an adaptive processor can be used for format conversion. Note that the transmitter has access to both the original and reconstructed pictures of the enhancement layer (with the use of the embedded base layer decoder in the encoder) and can make intelligent decisions about the adaptive signal processing. The adaptive format conversion information can then be transmitted as enhancement data. The second type of enhancement data can be created by encoding the residual, the difference between the decoded base layer after it has been converted to the format of the enhancement layer and the original enhancement layer video. Note that residual coding can be performed whether nonadaptive or adaptive format conversion is performed. The coded residual can be sent together with adaptive format conversion information in one enhancement layer or transmitted separately as another enhancement layer.

Figure 3.2 illustrates the corresponding receiver to the transmitter in Figure 3.1. The base layer bitstream can be decoded to create the base layer video. Note that the encoding and decoding of the base layer in a scalable coding scheme is identical to the

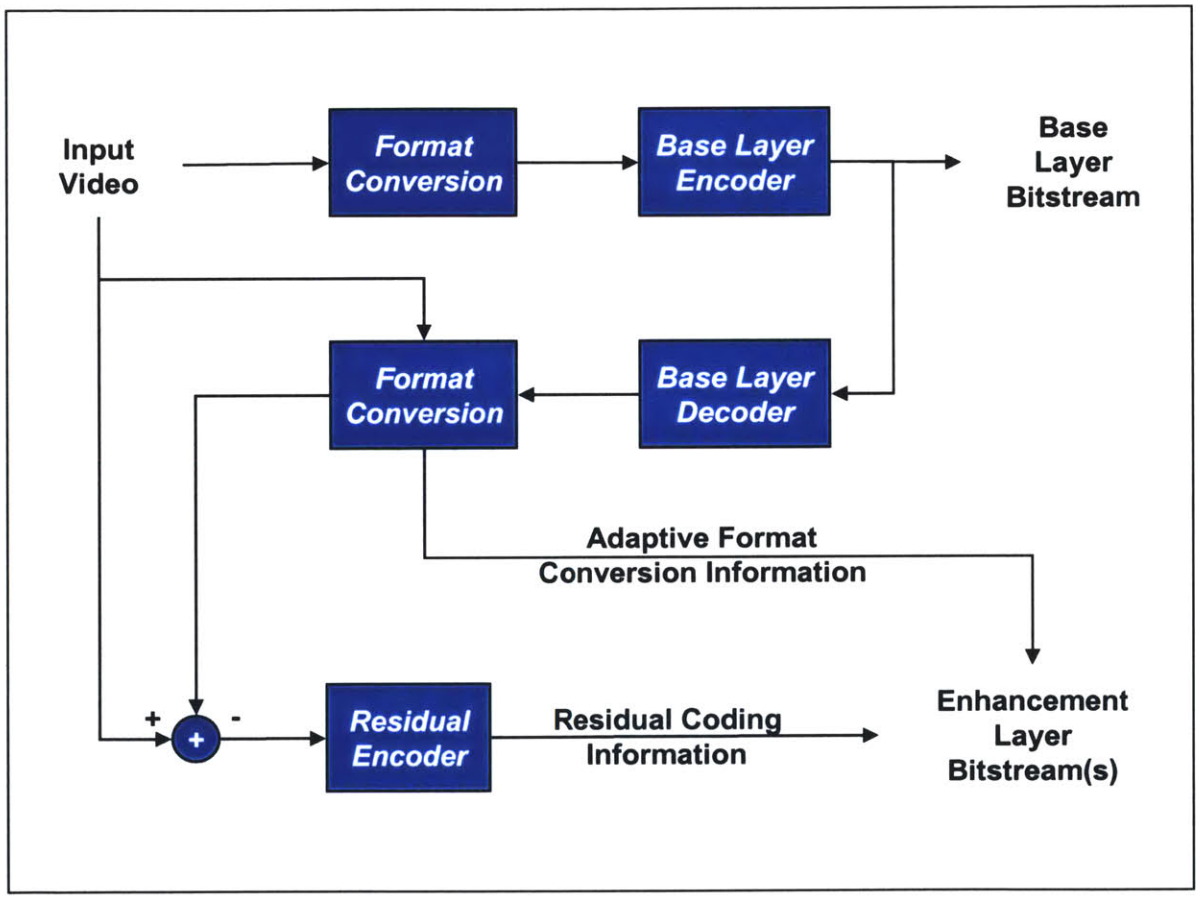

Figure 3.1: The Transmitter of a Scalable Coding System With Two Layers. Note that there are two possible types of enhancement data (adaptive format conversion information and residual coding information) and residual coding can be performed whether or not adaptive format conversion is performed.

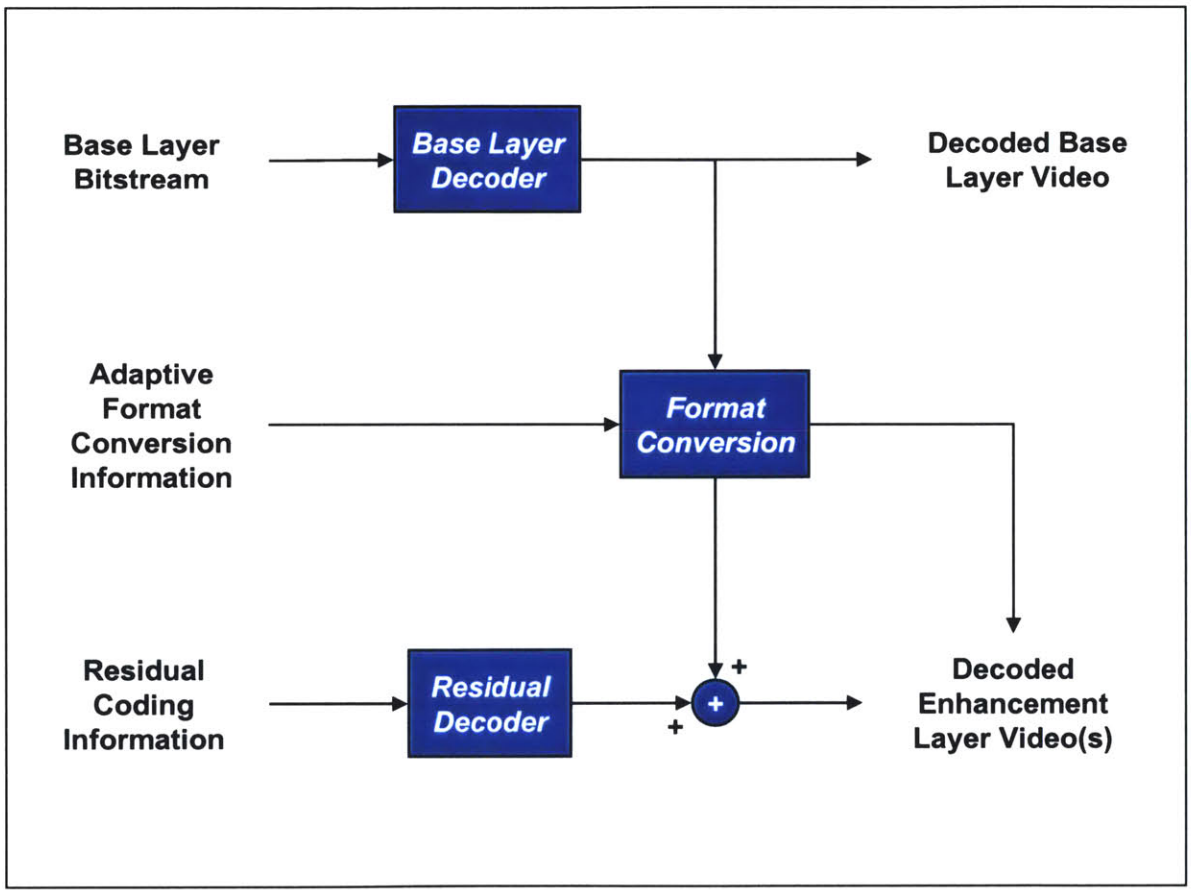

Figure 3.2: The Receiver of a Scalable Coding System With Two Layers. Base layer reception is identical to the single layer case, thus backward-compatibility is achieved.

single layer case. Therefore, no additional complexity beyond single layer decoding is required for reception of the base layer. This backward-compatibility is a very important property of scalable coding that allows scalable coding to be implemented seamlessly on top of existing systems. The decoded base layer can then be used with adaptive format conversion and/or residual coding information to create the enhancement layer video(s).

The characteristics of adaptive format conversion and residual coding are very different and choosing to utilize either or both types of enhancement information certainly depends on the available bandwidth. The different characteristics of both types of enhancement data will be analyzed and discussed in more detail later in this thesis, but it is useful to briefly mention some of the pros and cons of using adaptive format conversion information to suggest how adaptive format conversion may be useful when it is used instead of or in addition to residual coding.

The number of bits transmitted by using adaptive signal processing is small compared to residual coding. If the transmitter and receiver both have knowledge of the different signal processing techniques being used, an adaptive processor only needs to tell the receiver which type of processing to use and this should require only a few parameters per coding region. For example, if adaptive filtering is performed and the four different filters that are being used are known to both the transmitter and receiver, only two bits have to be sent to indicate the appropriate filter to use. On the other hand, a residual coder has to code a large number of quantized coefficients per region. A coarse quantizer could be used to limit the number of nonzero coefficients of the coded residual, but a coarsely quantized residual will usually not provide significant assistance with reconstruction of the video. Quantizers also have a limited scale and use of the coarsest quantizer may still generate excessive bits and cause the coded residual to exceed the available bandwidth. Therefore, adaptive format conversion may be the only type of enhancement information possible for low enhancement bitrates. The bitrate flexibility afforded by adaptive processing can be a major advantage, especially in coding scenarios where the enhancement layer bandwidth is small.

As stated earlier, this thesis addresses scalable video coding from the video coding point of view where the focus is on maximizing the coding efficiency and other issues such as codec complexity are considered secondary. However, it is important to state that the use of adaptive format conversion does increase the codec complexity and this may or may not be an issue depending on the particular application. The complexity of the

encoder is increased since it has to not only implement and decide between the different signal processing techniques, but the encoder must also code the additional side information. The complexity of a receiver is also increased since it must be able to implement the different methods or modes of format conversion, but note that the receiver does not have to decide which signal processing method to implement for each coding region since the bitstream will specify which mode to use. Therefore, the increase in receiver complexity is not as significant as in the encoder. Since complexity is a major determinant of cost, the use of adaptive format conversion would significantly increase the cost of an encoder and slightly increase the cost of a decoder. This agrees with a common economic model for many consumer applications which consists of a small number of expensive encoders and a large number of relatively cheap decoders.

3.2 Review of Previous Research

The concept of using adaptive format conversion in a scalable coding scheme has not been studied in great detail. Sunshine [7, 8] examined a special case of the general adaptive format conversion system: adaptive deinterlacing for a two service level system. In this system, the base layer was interlaced video and the enhancement layer was progressive video of the same spatial resolution. The main result of this work was that adaptive deinterlacing could significantly improve the decoded video quality of the enhancement layer (compared to nonadaptive deinterlacing of the base layer) with the transmission of a small amount of enhancement data. This result was interesting, especially since the improved quality was due solely to adaptive format conversion since no residual coding was performed. The target application for this research was the migration path for digital television. Closer inspection of the simulation results suggest issues that must be investigated further before adaptive deinterlacing can be applied to the migration path or another application. These issues include the lack of base layer coding and the relation between adaptive format conversion and residual coding.

The lack of base layer coding significantly affects the simulation results since the deinterlacing methods can utilize "perfect information" from the remaining fields to reconstruct the missing fields. Therefore, while results with an uncoded base layer are useful to establish empirical upper bounds on the performance of adaptive format con-

version, the results may not be applicable to many applications since most applications will require compression of the base layer and it is not clear how the quality of the base layer affects adaptive format conversion.

Residual coding is relatively well-understood and most existing scalable coding schemes utilize residual coding, so it is important to understand the differences between adaptive format conversion and residual coding to determine when and where adaptive format conversion can improve video scalability. As mentioned in the last section, intuition suggests that adaptive format conversion can provide scalability at low enhancement bitrates that are not possible with residual coding. However, the relation between the achievable rates and distortions of the two types of enhancement data must be investigated further. In addition, since adaptive format conversion can also be used with residual coding, it is useful to investigate whether adaptive format conversion can also assist video scalability at higher enhancement bitrates.

Additional issues with the implementation presented by Sunshine include a suboptimal algorithm to select the frame partitioning for adaptive block size experiments and unrealistic calculations of the enhancement bitrate due to the use of entropy codes. All of these issues will be addressed in the remainder of this chapter to develop a better implementation to evaluate the potential of adaptive format conversion for video scalability.

3.3 Implementation Overview

This thesis will choose to examine the same example of adaptive format conversion that Sunshine first investigated: adaptive deinterlacing. Other types of adaptive format conversion include adaptive spatial upsampling and adaptive temporal upsampling and can be investigated in a similar manner. This section provides an overview of the implementation used to obtain the simulations presented in this thesis. Two test sequences (Carphone and News) were examined using the implementation described below with interlaced video as the base layer and progressive video with the same number of lines as the enhancement layer. The original progressive scan sequences were Common Intermediate Format (CIF) resolution (288 x 352 pixels) and 30 frames long. The first frame from each of these test sequences is shown in Figure 3.3.

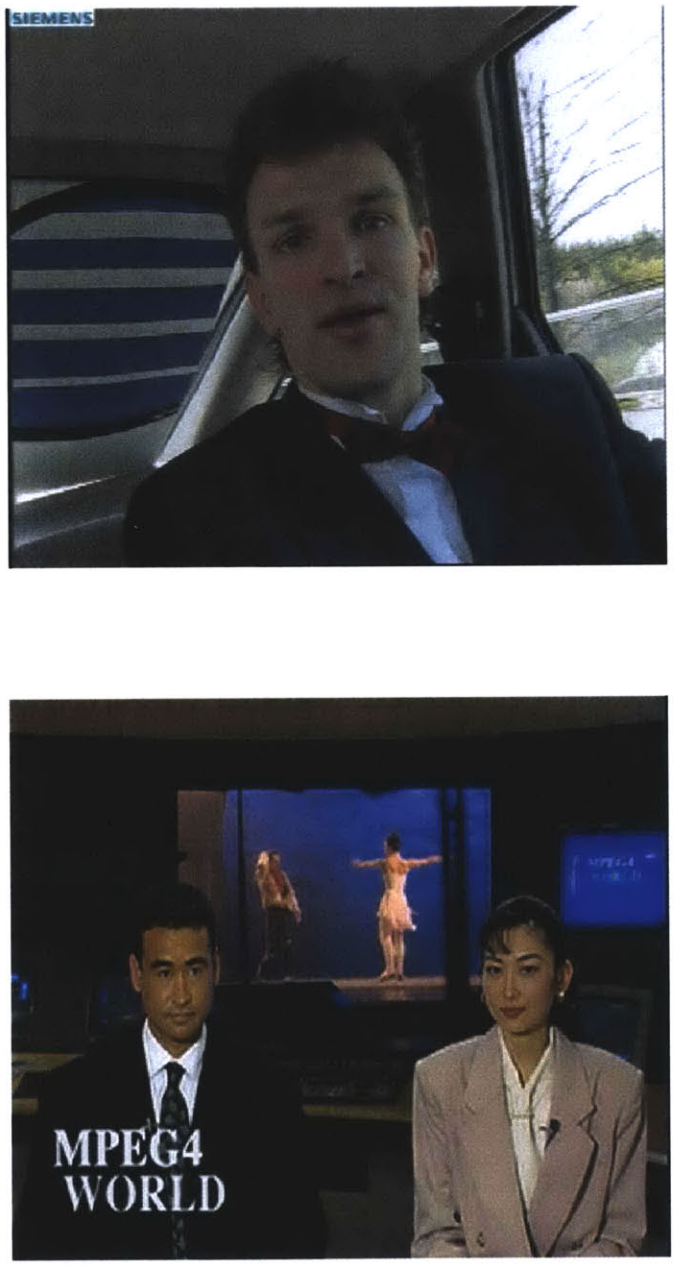

Figure 3.3: Test Sequences. A sample frame from the Carphone sequence (top) and the News sequence (bottom).

We will now define some notation to permit simpler discussion in the subsequent text. Let $F_P[x, y, n]$ represent a pixel of the original progressive sequence at the horizontal position x, the vertical position y and time n. Let $F_I[x, y, n]$ represent the corresponding interlaced sequence. Since an interlaced sequence consists of alternating fields, $F_I[x, y, n]$ is not defined when $mod(y, 2) \neq mod(n, 2)$ where the modulus operator is defined as

$$mod(a, 2) = \begin{cases} 0, & a \text{ even} \\ 1, & a \text{ odd} \end{cases} \tag{3.1}$$

Note that this definition follows from the conventions that the frames and rows are enumerated starting from zero and the even field is the first coded field for an interlaced sequence. The progressive enhancement layer was converted to the interlaced base layer by simply extracting the appropriate fields of the progressive video. Therefore,

$$F_I[x, y, n] = \begin{cases} F_P[x, y, n], & mod(y, 2) = mod(n, 2) \\ \emptyset, & mod(y, 2) \neq mod(n, 2) \end{cases} \tag{3.2}$$

The interlaced base layer is then coded as described in Section 3.3.1. Let $\hat{F}_I[x, y, n]$ represent the decoded interlaced sequence. Adaptive deinterlacing and/or residual coding are then used to create the decoded progressive enhancement layer, $\hat{F}_P[x, y, n]$. The adaptive deinterlacing implementation is presented in Section 3.3.2 and the residual coding implementation is described in Section 3.3.3.

The measure of decoded video quality is the Peak Signal-to-Noise Ratio (PSNR) of the luminance component between the original and decoded video. Therefore, $F_I[x, y, n]$ and $\hat{F}_I[x, y, n]$ are used to compute the PSNR of the base layer. Similarly, $P[x, y, n]$ and $\hat{F}_P[x, y, n]$ are used to compute the PSNR of the enhancement layer.

Although the luminance component is the only component used by the PSNR distortion metric, all three components of the 4:2:0 YUV sequence are coded when computing the bitrate which is expressed in Bits Per Pixel (BPP).

3.3.1 Base Layer Coding

The most significant issue with the previous work on adaptive deinterlacing is the use of an uncoded base layer, i.e. no compression was applied to the interlaced base layer before it was adaptively deinterlaced to create the progressive enhancement layer. This is very significant because it allows the deinterlacing modes to use perfect information about the available field(s) to reconstruct the missing field. Quantization effects from compression will not only degrade the base layer, but also degrade the enhancement layer since the deinterlacing utilizes information from the base layer. Therefore, results obtained with an uncoded base layer can be considered an upper bound on the performance of adaptive format conversion since compression effects are not present. Upper bound calculations are useful; however, simulations with a coded base layer must be performed to better understand adaptive deinterlacing and for the results to have practical importance.

A base layer codec was included in the simulations performed in this thesis to investigate the effect of the base layer. The base layer was coded with an MPEG-2 encoder using only Intra-frames with the quantization parameter fixed to one value for the whole sequence. In addition to having a simple implementation, fixed quantization encoding was chosen to eliminate any spurious effects from rate control. The quantization parameter was varied ($Q = 2, 6, ..., 30$ and $38, 46, ..., 62$) to provide a wide range of base layer quality. Note that no attempt was made to optimize the coding efficiency of the base layer. Base layer coding is included in these simulations to investigate the effect of the base layer on scalable coding. Since the rate-distortion characteristics of the base layer can be considered to be a one-to-one function, this thesis will choose to utilize the base layer distortion for its analysis. The advantage of this approach is that it eliminates the coding efficiency of the base layer coder from the analysis. The disadvantage of this approach is that the base layer bitrate has limited quantitative value. The subtle distinction of using the base layer distortion instead of the base layer bitrate will be discussed in more detail in the problem formulation section (Section 4.1) of the next chapter.

3.3.2 Adaptive Deinterlacing

The implementation of adaptive deinterlacing in the scalable codec is separated into three sections discussing the frame partitioning, the deinterlacing modes and the parameter

coding.

3.3.2.1 Frame Partitioning

Two different frame partitioning schemes were examined. The first partitioning scheme had nonoverlapping blocks of the same size. Three different block sizes were examined: 16 x 16 pixels, 8 x 8 pixels and 4 x 4 pixels. The second partitioning scheme allowed adaptive block sizes and was initialized by first dividing each frame into nonoverlapping 16 x 16 blocks. Each of these 16 x 16 blocks could be divided into four 8 x 8 blocks, each of which could be further subdivided into four 4 x 4 blocks. Figure 3.4 illustrates the possible frame partitionings for one 16 x 16 block when it is divided into nonoverlapping blocks of size 16 x 16, 8 x 8 and 4 x 4. For ease of discussion, it is useful to define a convention for ordering the blocks for each partitioning. If the block consists of a single 16 x 16 block, there is no ordering required. Otherwise, examine the four nonoverlapping 8 x 8 blocks in raster scan order (from left to right and top to bottom) and for each block, determine if they are to be coded as a single 8 x 8 block or subdivided into four 4 x 4 blocks. For the former case, the 8 x 8 block is simply the next block in the ordering and for the latter case, scan the four 4 x 4 blocks in raster scan order. The numbers inside each block indicate the scanning order of that block when this convention is followed.

Table 3.1 shows the six different frame partitionings along with the number of each type of block, the number of different permutations and the number of modes that need to be coded for that particular partitioning. For example, the third line in Table 3.1 states that there are four permutations and seven modes when a 16 x 16 block is divided into three 8 x 8 blocks and four 4 x 4 blocks. Figure 3.5 shows the four different possible permutations with three 8 x 8 blocks and four 4 x 4 blocks. Similarly, Figure 3.6 shows the six different possible permutations with two 8 x 8 blocks and eight 4 x 4 blocks and Figure 3.7 shows the four different possible permutations with one 8 x 8 blocks and twelve 4 x 4 blocks. As in Figure 3.4, the numbers inside each block inside the scanning order of that block.

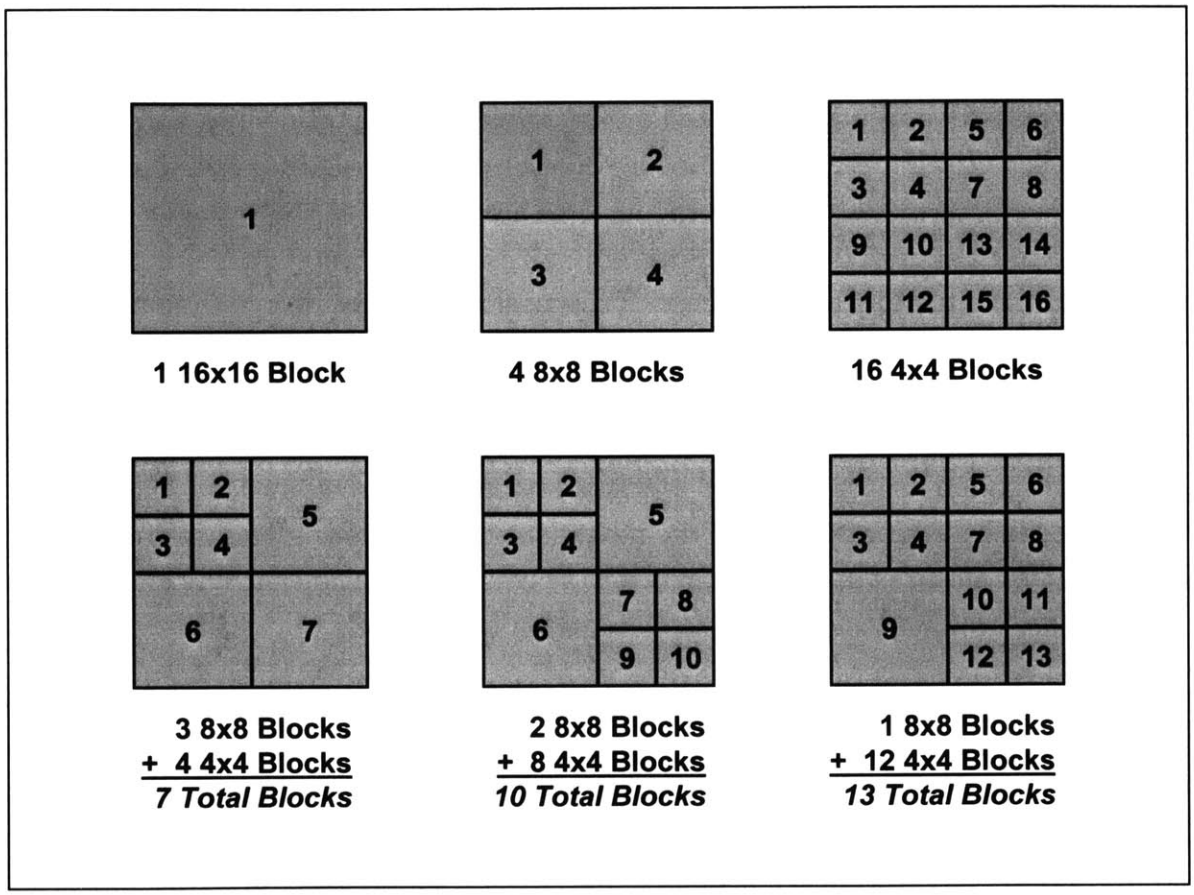

Figure 3.4: Frame Partitionings. The figures illustrate the different frame partitionings for one 16 x 16 block with nonoverlapping blocks of size 16 x 16, 8 x 8 and 4 x 4. The numbers inside each block indicate the scanning order of that block when the convention described in the text is followed.

Number of 16 x 16 Blocks	Number of 8 x 8 Blocks	Number of 4 x 4 Blocks	Different Permutations	Number of Modes
1	0	0	1	1
0	4	0	1	4
0	3	4	4	7
0	2	8	6	10
0	1	12	4	13
0	0	16	1	16

Table 3.1: Blocks, Permutations and Modes of the Frame Partitionings. Each row represents a different manner in which a 16 x 16 block can be partitioned. In addition to the number of each type of block, the table also lists the number of permutations and the number of modes that must be coded.

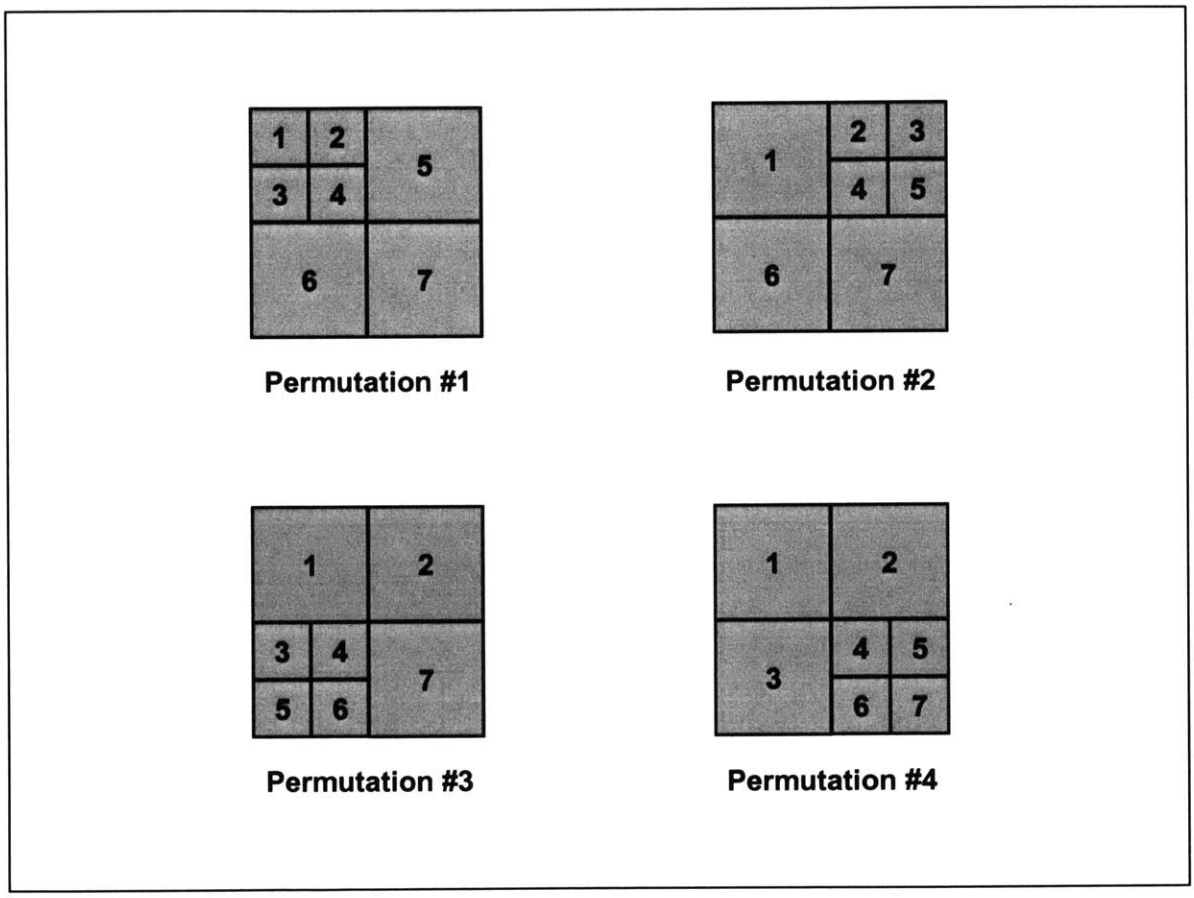

Figure 3.5: The Four Possible Permutations When One 16 x 16 Block is Partitioned Into Three 8 x 8 Blocks and Four 4 x 4 Blocks. The numbers inside each block indicate the scanning order of that block.

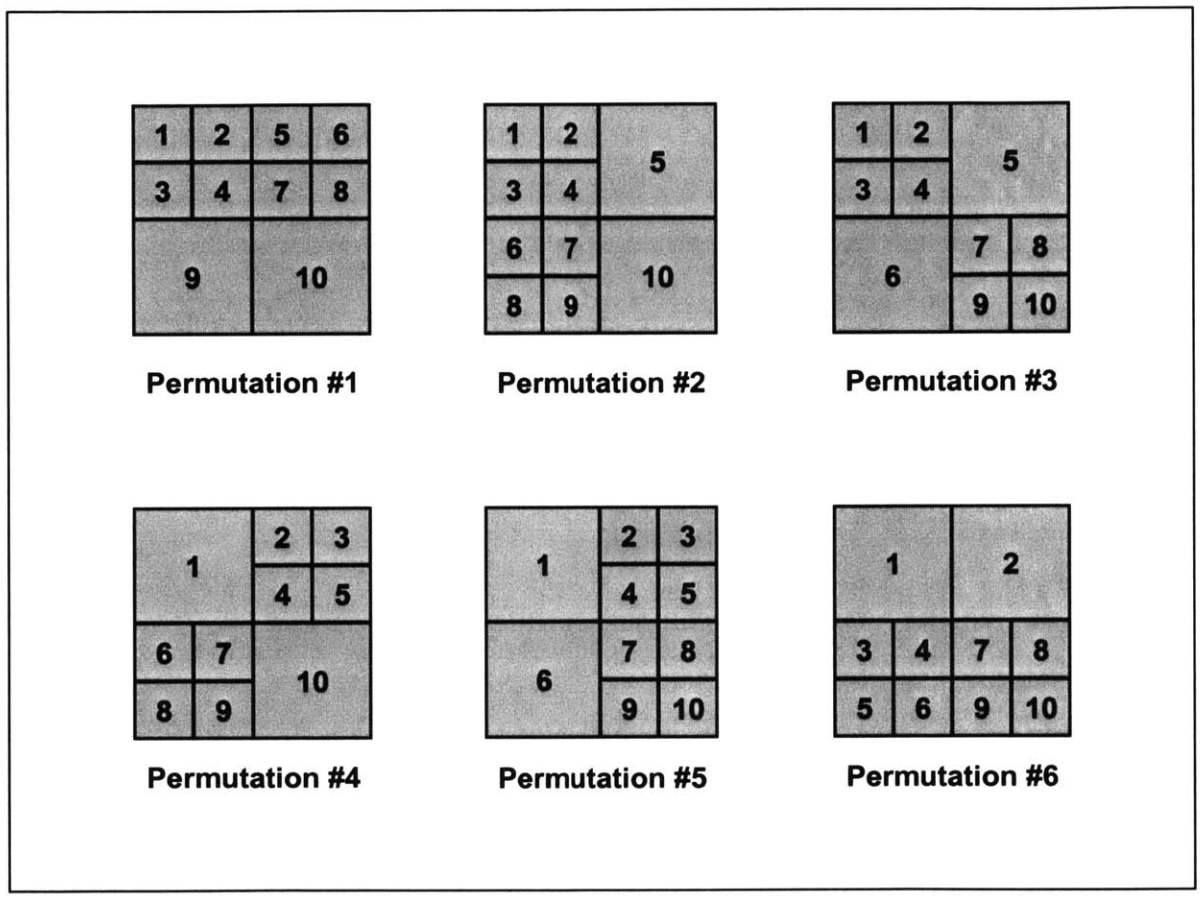

Figure 3.6: The Six Possible Permutations When One 16 x 16 Block is Partitioned Into Two 8 x 8 Blocks and Eight 4 x 4 Blocks. The numbers inside each block indicate the scanning order of that block.

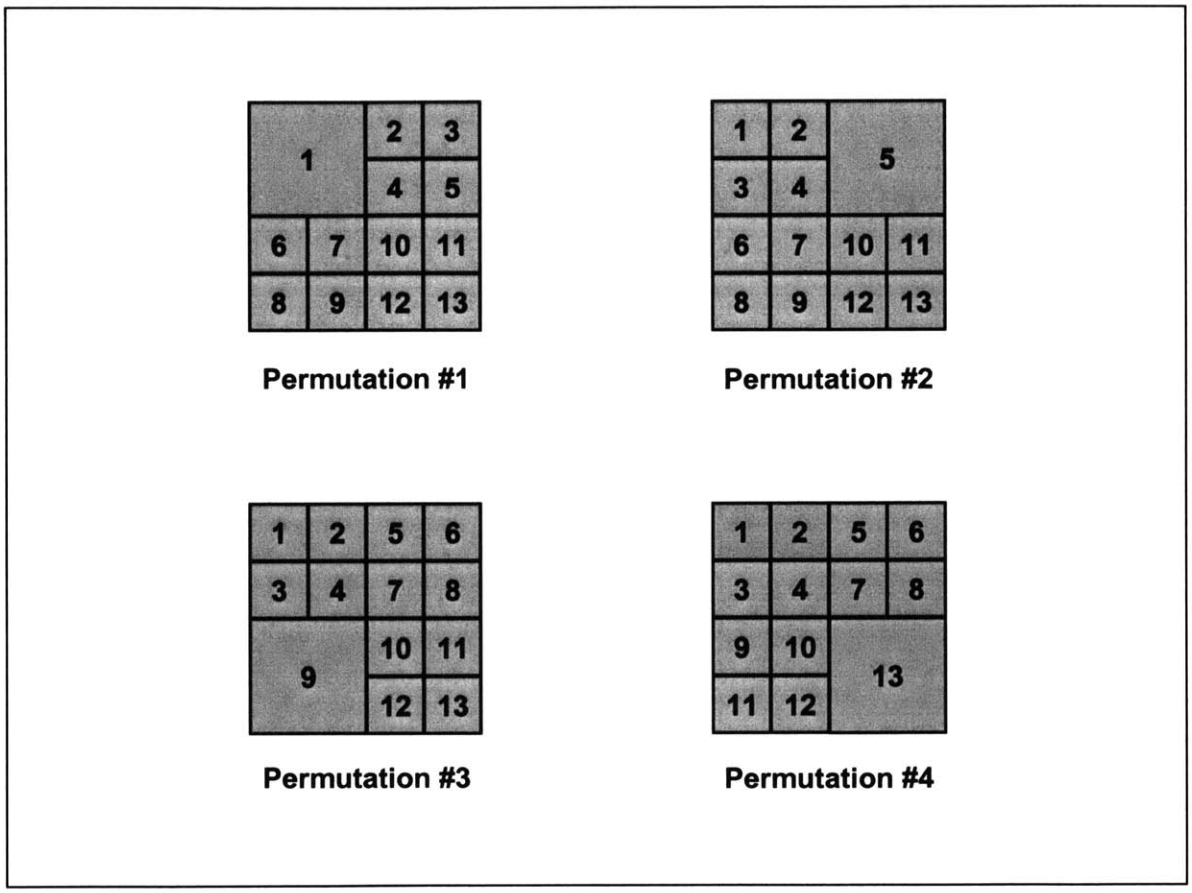

Figure 3.7: The Four Possible Permutations When One 16 x 16 Block is Partitioned Into One 8 x 8 Block and Twelve 4 x 4 Blocks. The numbers inside each block indicate the scanning order of that block.

Adapting the block size allows the encoder to concentrate bits where they are needed most and also provides finer levels of bitstream scalability than a partitioning scheme with fixed block sizes. The disadvantage of adaptive block sizes is the additional overhead that needs to be transmitted to represent the frame partitioning. Note that the number of subdivisions within a 16 x 16 block will be limited by the enhancement layer bandwidth since each subdivision would increase the number of deinterlacing modes that would need to be transmitted.

3.3.2.2 Deinterlacing Modes

Deinterlacing is a topic that has been studied extensively in the literature [25] and many algorithms have been proposed to reconstruct the original progressive video by attempting to determine characteristics of the progressive video from the corresponding interlaced video without transmitting additional side information and/or access to the original source. Interest in deinterlacing has recently increased due to progressive displays such as computer monitors becoming more prevalent. Since there is a large amount of video equipment and content in the interlaced format (due to the existing analog standard), deinterlacing is required to display this material on a progressive display. Deinterlacing methods vary greatly in complexity and performance, but can often be classified into two types: intraframe methods and interframe methods.

Intraframe methods only use pixels from the interlaced sequence that are in the same field to reconstruct the missing lines. Many intraframe methods have been developed to exploit the spatial redundancy between pixels in the known and missing lines. Since intraframe deinterlacing methods only utilize pixels in the current frame, these methods do not require additional video storage to implement. Intraframe methods are also very robust to the motion in a video sequence since they only consider a single frame at a time. The tradeoff for this robustness is the inability of intraframe methods to exploit the large temporal correlation that is often found in successive fields of an interlaced sequence.

Interframe methods use pixels in previous and/or subsequent fields to reconstruct missing lines in the current field. These methods are often very effective at reconstructing temporally invariant regions such as static backgrounds. Interframe methods require storage of one or more fields for implementation. Video storage was expensive and gen-

erally infeasible for many applications in the past, but the reduced cost of memory has make video storage a viable option in current applications.

It is often difficult to choose between using intraframe and interframe information for deinterlacing. There are hybrid methods that utilize both types of information, but the fundamental issue for efficient deinterlacing is to determine which type of information to use. This "blind" deinterlacing problem can be a very difficult problem as shown in Figure 3.8. In this example, one would like to reconstruct a pixel from the progressive sequence $\hat{F}_P[x, y, n]$ which lies on a missing line. The location of this unknown pixel is shown by the "?" in the figure. The circles represent some of the known pixels from the corresponding interlaced sequence $\hat{F}_I[x, y, n]$. Note that other pixels could also be used but this subset of known pixels is sufficient for this discussion. The interlaced sequence provides both intraframe information (such as the pixels on lines $y - 1$ and $y + 1$ in field n) and interframe information (such as the pixels on line y in fields $n - 1$ and $n + 1$), but one cannot make any general claims on how to properly utilize this information for reconstructing $\hat{F}_P[x, y, n]$.

It is important to note that most of the previous work on deinterlacing was performed on interlaced video where an original progressive source was not available. An example of this scenario is video capture with an interlaced camera that is transmitted and then displayed on a progressive display such as a computer monitor. In this case, there is no original source to assist the deinterlacing, so any deinterlacing performed is done to create the best visual effect at the receiver which is not the same as attempting to reconstruct the original video seen by the capture device. Adaptive deinterlacing is different from almost all of the deinterlacing problems investigated in the past because the original progressive video is available to the encoder and the encoder can utilize the original to determine and transmit additional side information to assist the deinterlacing at the receiver. Thus, deinterlacing decisions at the receiver do not have to be "blind" and can be done in an intelligent fashion with the additional side information. One reason that adaptive deinterlacing was chosen as the specific example of adaptive format conversion to be investigated in this thesis (instead of adaptive spatial or temporal upsampling) is that the encoder can accurately determine whether the deinterlacing should use intraframe or interframe information with access to the original source. Therefore, the additional side information should significantly improve deinterlacing. Conceptually, adaptive spatial or temporal signal processing could have a similar effect as adaptive deinterlacing, but

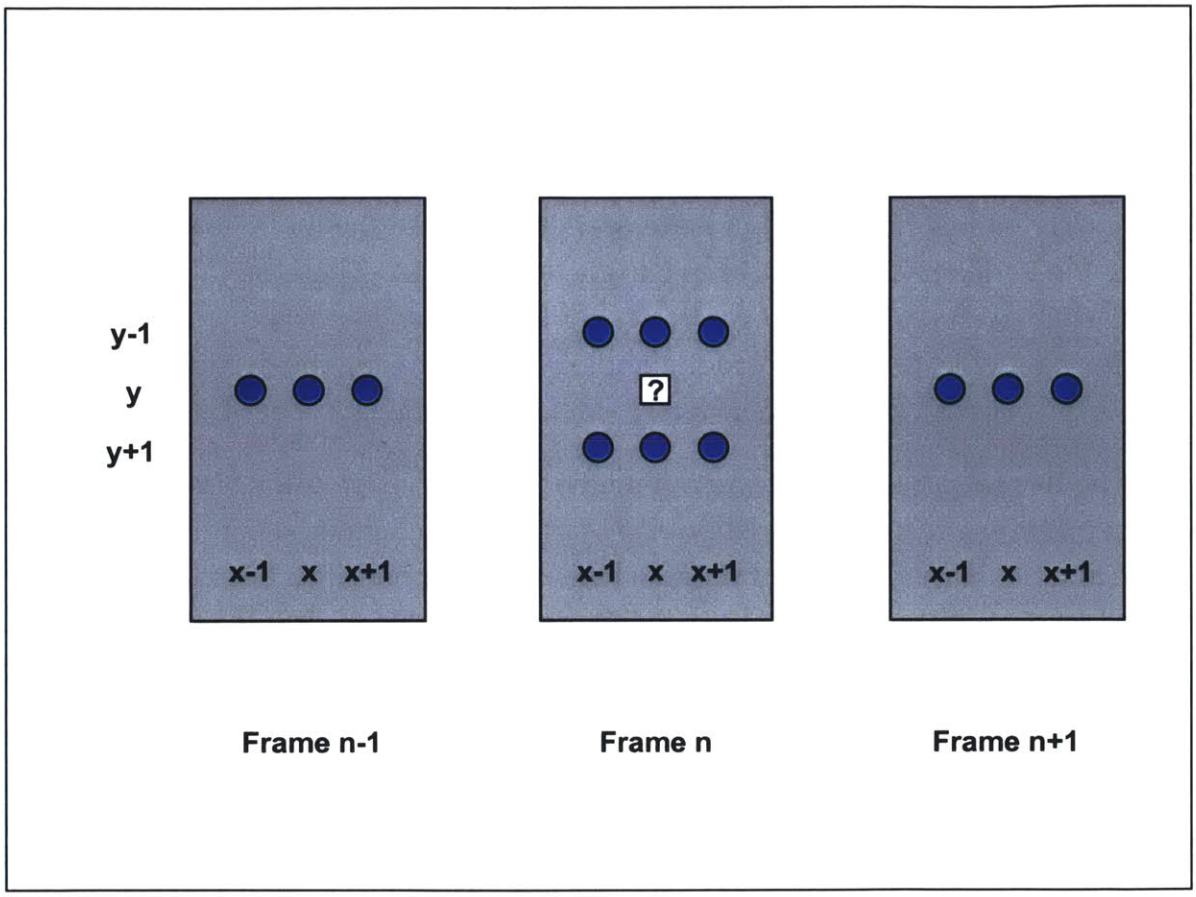

Figure 3.8: Intraframe or Interframe Information? The example shows how both intraframe and interframe information could be used to reconstruct $\hat{F}_P[x, y, n]$. The circles represent known pixels from the corresponding interlaced sequence $\hat{F}_I[x, y, n]$.

the coding gain would probably not be as significant.

Since only the best mode is chosen for each region, it is more important to choose modes that handle certain regions very well (and may handle other regions poorly) rather than selecting modes that perform adequately over all regions. Sunshine stated that a small number of simple deinterlacing modes would be sufficient to handle different video attributes such as spatial correlation, stationary regions and translational motion. The implementation constructed for the simulations in this thesis has four different deinterlacing modes: linear interpolation, Martinez-Lim deinterlacing, forward field repetition and backward field repetition. These simple modes were chosen because of their adequate performance and relative ease of implementation. Note that other deinterlacing modes can be chosen which may result in better performance but the main goal of this work is to prove the basic concepts of adaptive format conversion. Therefore, the use of simple deinterlacing modes is sufficient for this analysis. It is also important to note that simple deinterlacing modes reduce the codec complexity. Minimizing the decoder complexity is desirable for many applications since there are often a large number of decoders.

The two intraframe deinterlacing methods used to exploit the spatial redundancy between pixels are linear interpolation and Martinez-Lim deinterlacing. Linear interpolation is a simple deinterlacing technique where a missing line is reconstructed by averaging the lines directly below and above it and is shown in Figure 3.9. If a line only has one neighbor, that line is simply copied to create the missing line, i.e. line repetition. Therefore, if $\hat{F}_I[x,y,n]$ and $\hat{F}_P[x,y,n]$ represent the interlaced and deinterlaced (progressive) sequences, respectively, the formal definition of linear interpolation is:

$$\hat{F}_P[x,y,n] = \begin{cases} \hat{F}_I[x,y,n], & mod(y,2) = mod(n,2) \\ \frac{\hat{F}_I[x,y-1,n]+\hat{F}_I[x,y+1,n]}{2}, & mod(y,2) \neq mod(n,2) \end{cases} \tag{3.3}$$

Martinez-Lim deinterlacing [26] is a more sophisticated intraframe deinterlacing technique and is shown in Figure 3.10. Martinez-Lim deinterlacing begins with a parametric model that attempts to model the local region around the missing pixel. In this implementation, the five samples on the lines immediately above and below the missing line were fit to the following set of two dimensional second order polynomials:

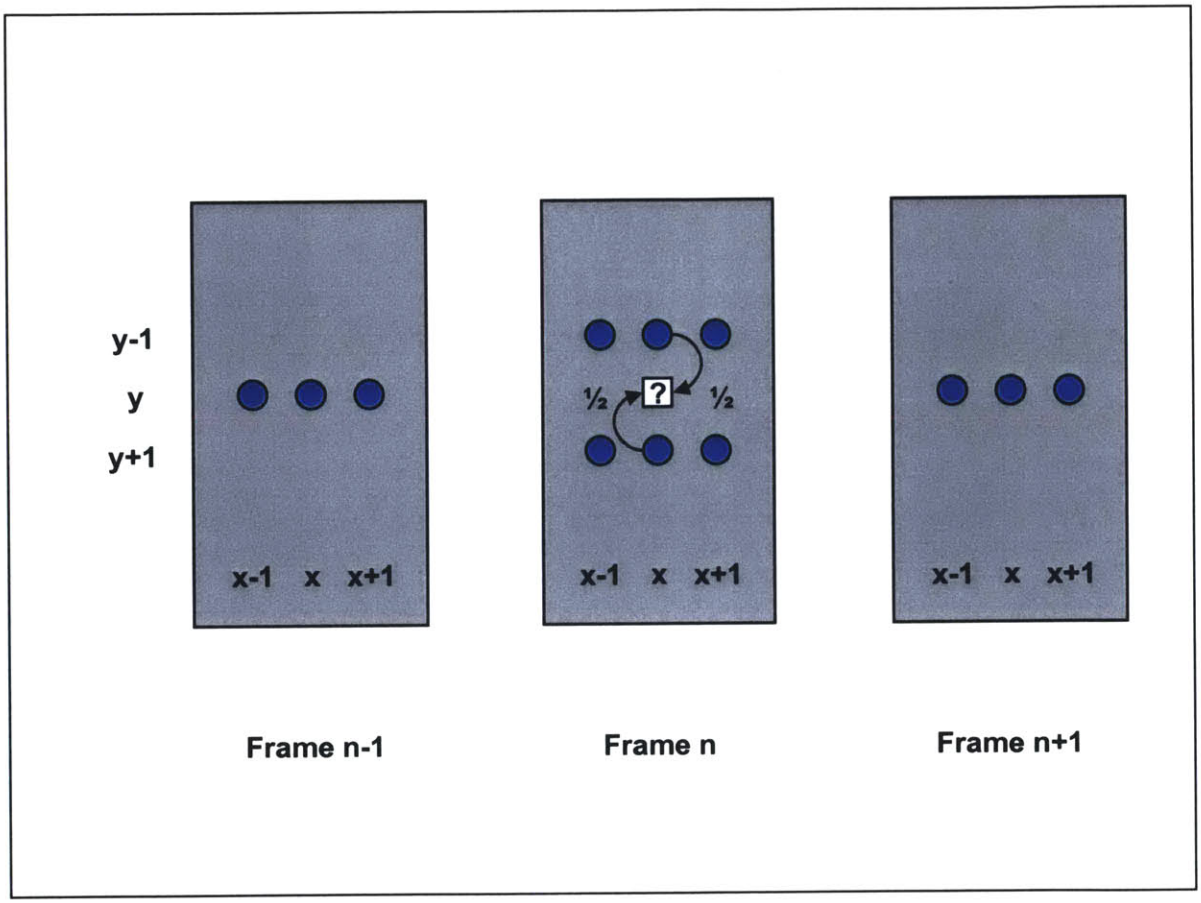

Figure 3.9: Linear Interpolation. The pixels directly above and below the missing pixel are averaged to reconstruct the missing pixel value.

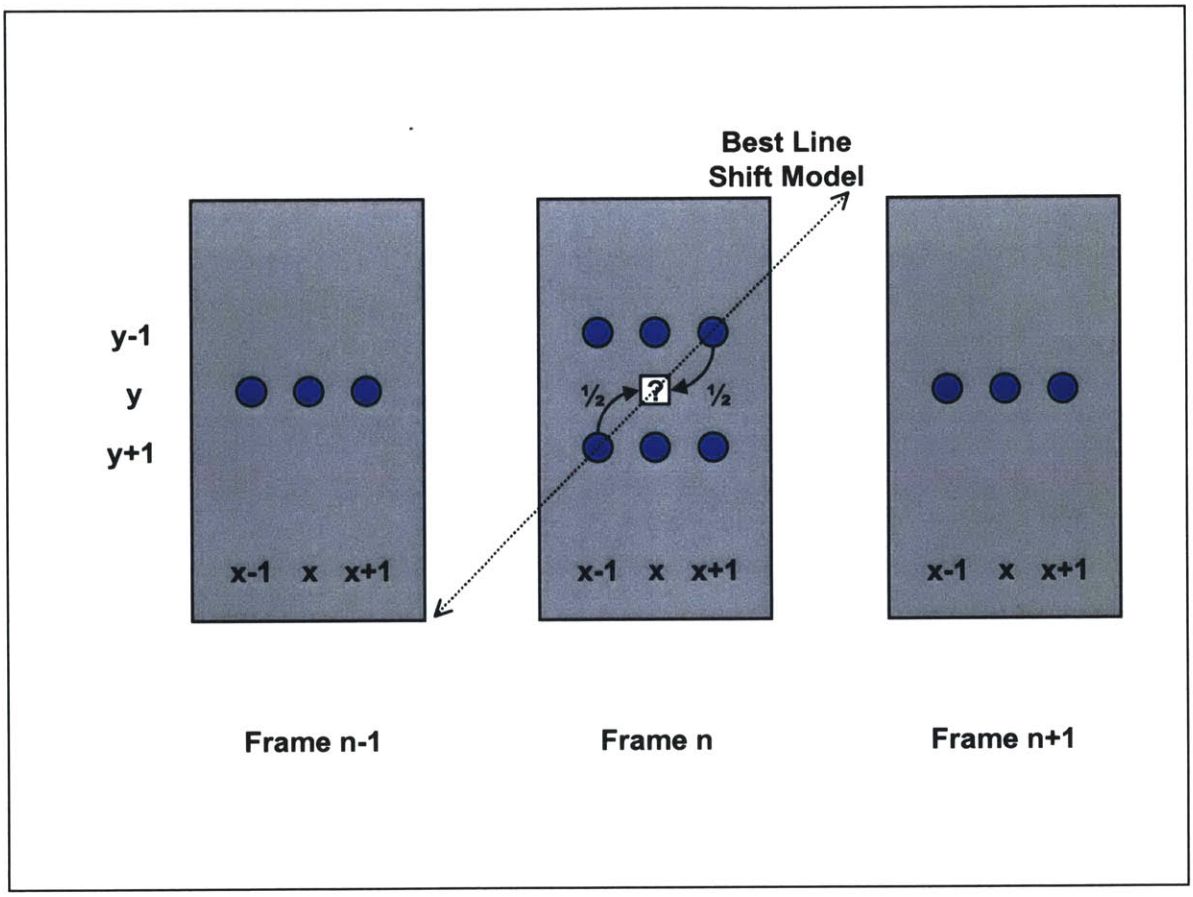

Figure 3.10: Martinez-Lim Deinterlacing. The local region around the missing pixel is fitted to a parametric model. The best line shift model is then computed and the pixels on the lines above and below the missing pixel that correspond to the line shift model are averaged to reconstruct the missing pixel value.

- $f(x,y) = 1$
- $f(x,y) = x$
- $f(x,y) = y$
- $f(x,y) = x^2$
- $f(x,y) = xy$

This model is then spatially interpolated to reconstruct the missing line by assuming a simple line shift model where small segments of adjacent scan lines are related by a simple horizontal shift. As in the linear interpolation implementation, lines with only one neighbor were reconstructed by copying the neighboring line to create the missing line, i.e. line repetition. Let x_o represent the (integer) horizontal shift corresponding to the best line shift model. The formal definition of Martinez-Lim deinterlacing is defined to be:

$$\hat{F}_P[x,y,n] = \begin{cases} \hat{F}_I[x,y,n], & mod(y,2) = mod(n,2) \\ \frac{\hat{F}_I[x+x_o,y-1,n]+\hat{F}_I[x-x_o,y+1,n]}{2}, & mod(y,2) \neq mod(n,2) \end{cases} \tag{3.4}$$

An advantage of deinterlacing techniques that incorporate image modeling such as Martinez-Lim deinterlacing is that they are less susceptible to noise. Since the model and not the actual image is spatially interpolated to reconstruct the missing line, noise that does not fit into the model ends up being disregarded. A disadvantage of this technique is the complexity required is greater than other deinterlacing techniques such as the simple linear interpolation method.

The two interframe deinterlacing modes used to reconstruct stationary regions were forward field repetition and backward field repetition. Forward field repetition and backward field repetition simply copy the corresponding lines from the previous and subsequent fields, respectively. Forward field repetition is shown in Figure 3.11 and defined to be:

$$\hat{F}_P[x,y,n] = \begin{cases} \hat{F}_I[x,y,n], & mod(y,2) = mod(n,2) \\ \hat{F}_I[x,y,n-1], & mod(y,2) \neq mod(n,2) \end{cases} \tag{3.5}$$

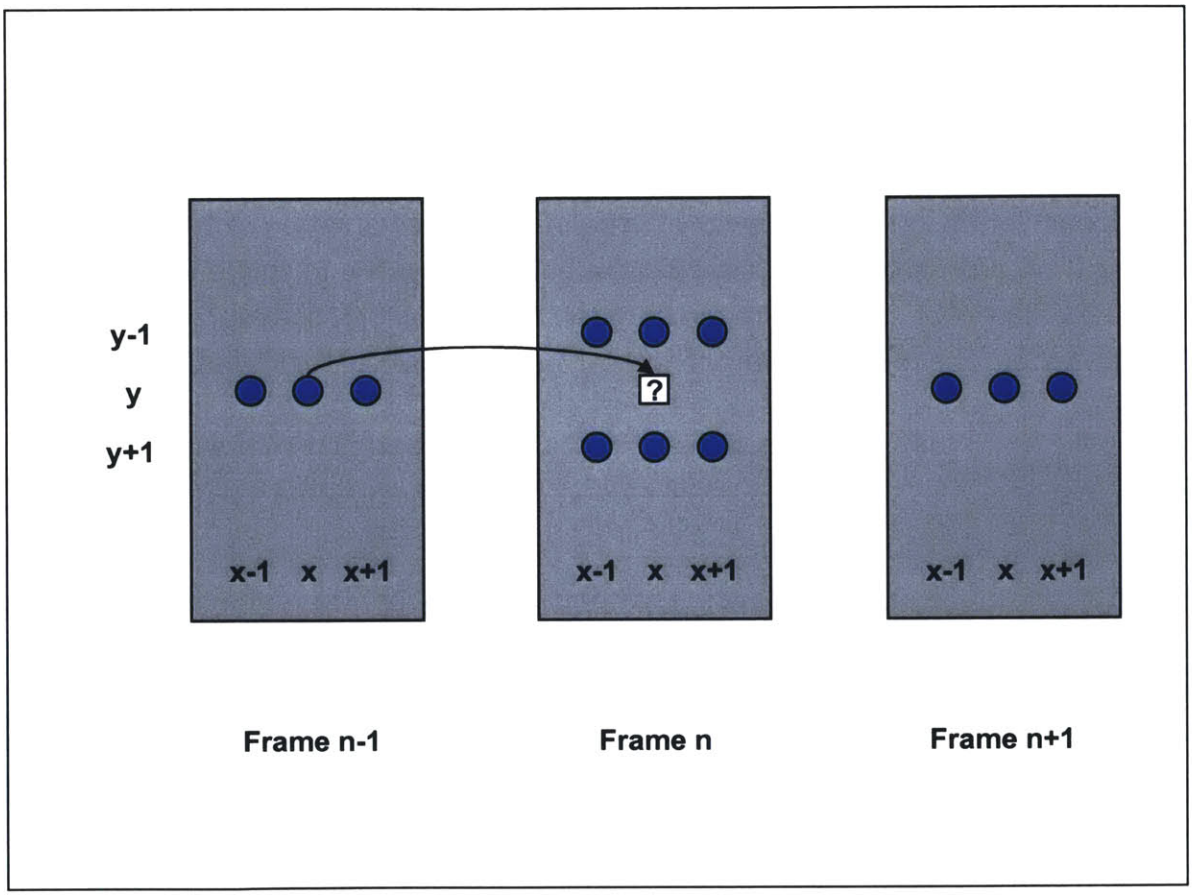

Figure 3.11: Forward Field Repetition. The pixel from the same spatial location of the previous field is used to reconstruct the missing pixel value.

Similarly, backward field repetition is shown in Figure 3.12 and defined to be:

$$\hat{F}_P[x,y,n] = \begin{cases} \hat{F}_I[x,y,n], & mod(y,2) = mod(n,2) \\ \hat{F}_I[x,y,n+1], & mod(y,2) \neq mod(n,2) \end{cases} \quad (3.6)$$

All of the deinterlacing modes used in this thesis reconstruct missing lines using information from the interlaced base layer. Conceptually, one could also use information from previously decoded frames of the enhancement layer and this subtle distinction may seem insignificant since previously decoded frames of the enhancement layer are also created from the base layer. However, deinterlacing modes that use previously decoded frames of the enhancement layer create a recursive structure that significantly complicates mode selection for adaptive format conversion. Figure 3.13 shows the fields of the base layer and the frames of the enhancement layer at times t_{n-1}, t_n and t_{n+1}. It is assumed that the interlaced base layer has already been coded and the goal is to reconstruct the progressive enhancement layer at time t_n. Deinterlacing modes can easily utilize any of the fields from the previously coded base layer which are represented by the black arrows. The temporally adjacent frames in the enhancement layer are represented by blue arrows. If these frames are used, it creates a significant dependency between frames in the enhancement layer. The complexity of the mode selection is then of an exponential order. However, if deinterlacing modes are limited to using only information from the base layer, there is no dependency between frames in the enhancement layer and decisions can be made on a frame-by-frame basis.

3.3.2.3 Parameter Coding

Sunshine chose to use entropy codes to measure the enhancement layer bandwidth required for the adaptive deinterlacing information. The use of entropy can be a good measure of the bit requirement to encode the enhancement information with the assumption that the bitstream can be characterized easily or that adaptive coding will quickly converge to the true statistics. This implementation uses variable length codes to provide results that would be more representative of a realistic implementation of adaptive deinterlacing.

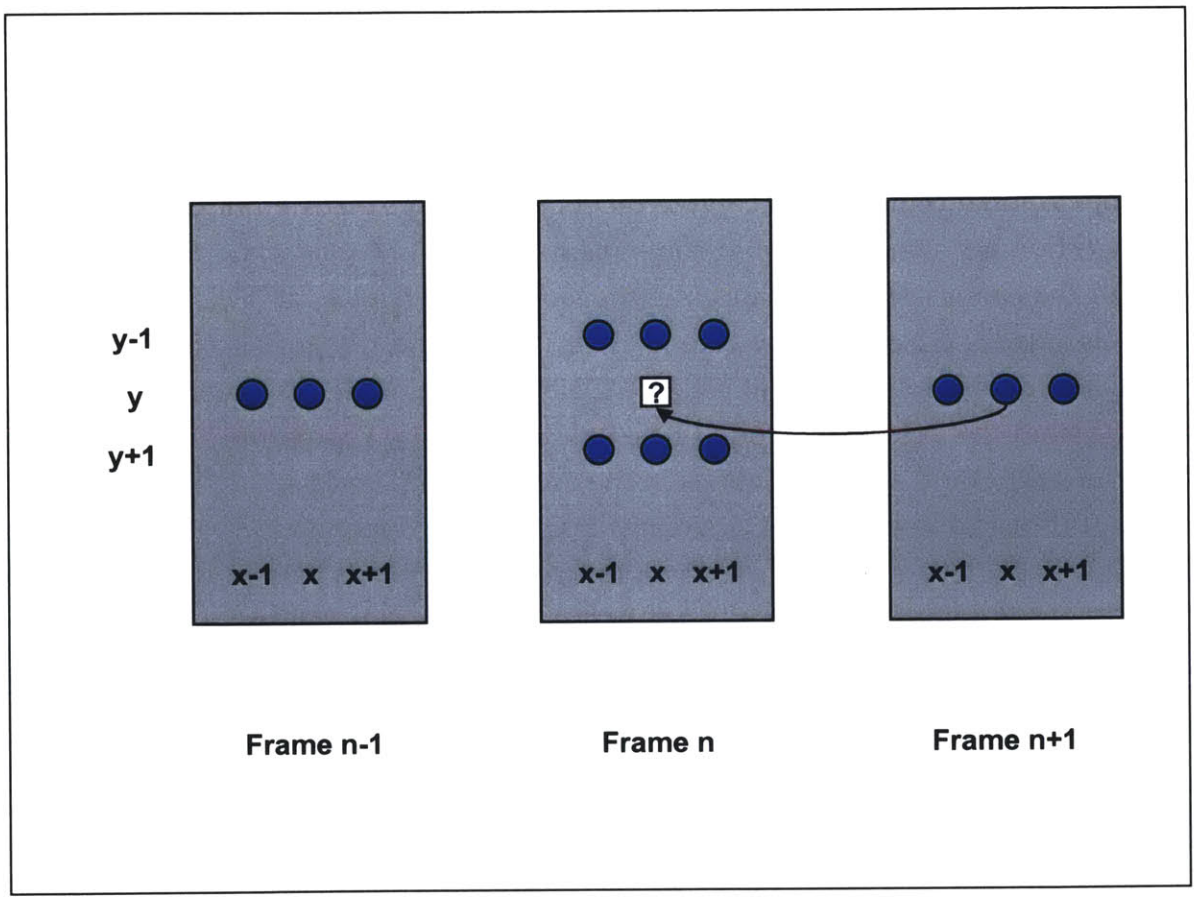

Figure 3.12: Backward Field Repetition. The pixel from the same spatial location of the subsequent field is used to reconstruct the missing pixel value.

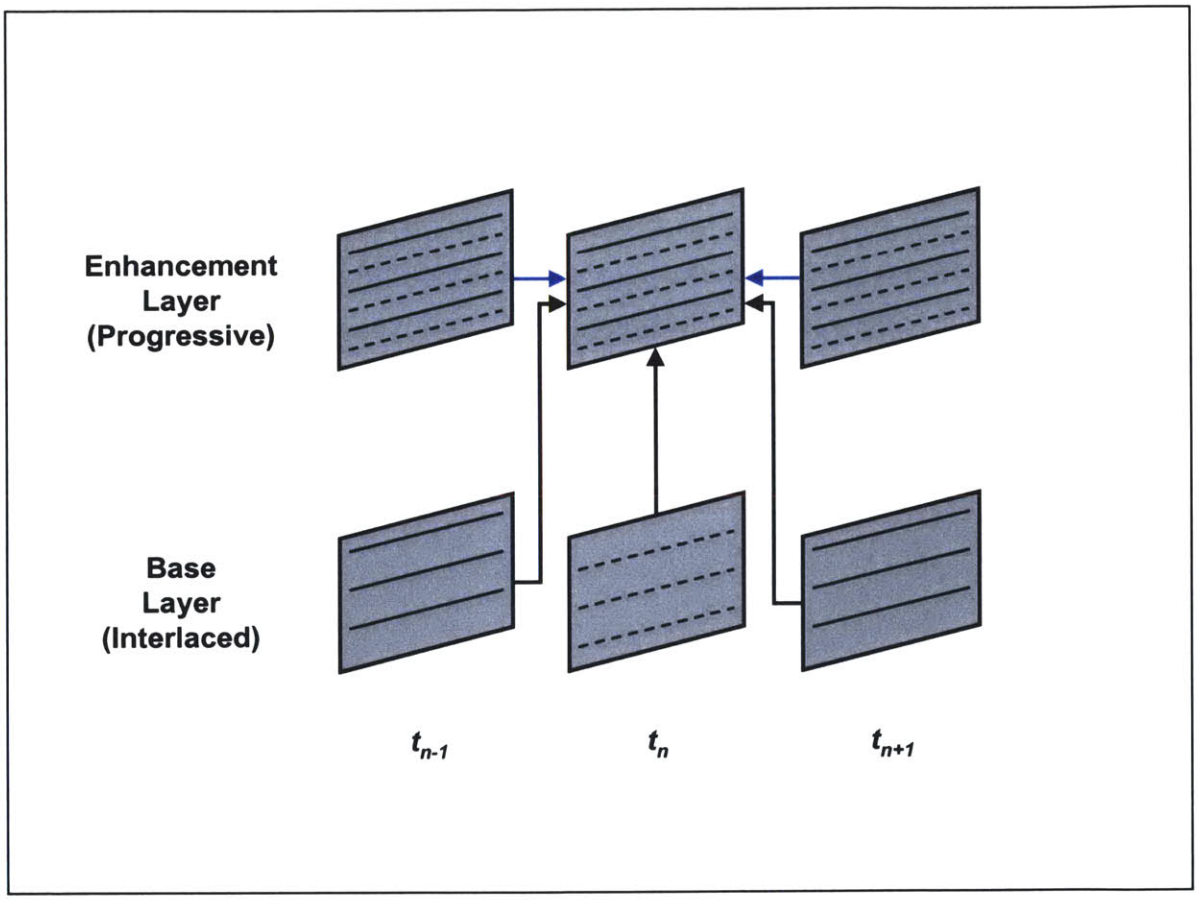

Figure 3.13: Information From the Base Layer. Reconstruction of frame t_n of the enhancement layer may utilize fields t_{n-1}, t_n and t_{n+1} of the base layer (black arrows) as well as frames t_{n-1} and t_{n+1} of the enhancement layer (blue arrows) if these frames were previously decoded. However, use of previously decoded frames in the enhancement layer causes deinterlacing decisions to be dependent across different frames and significantly increases the complexity of mode selection.

The enhancement bitrate required to transmit the deinterlacing modes for adaptive processing was computed by first computing the probability that each deinterlacing mode was used in each frame. A Huffman code was then created using these *a posteriori* probabilities. The order that the deinterlacing modes were coded was established by dividing each frame into nonoverlapping 16 x 16 blocks and coding these blocks in raster scan order. The modes within each 16 x 16 block were coded in the order shown in the Figures 3.4-3.7.

For adaptive deinterlacing with adaptive block sizes, the frame partitioning must also be transmitted and this bitrate was also computed using *a posteriori* probabilities. Note that there are 17 different ways to partition a 16 x 16 block in this implementation. Huffman codes were created using the *a posteriori* probabilities that each of these 17 different partitions were used and the partitions of the nonoverlapping 16 x 16 blocks were coded in raster scan order.

3.3.3 Residual Coding of the Enhancement Layer

The use of residual coding is well-known and used in most scalable coding schemes. Since adaptive format conversion can be used instead of or in addition to residual coding, it is important to examine both of these scenarios to better understand how adaptive format conversion can improve video scalability. A residual coder was included in this implementation to permit either or both types of enhancement data to be used. After the decoded base layer is converted to the enhancement layer format, the difference between this sequence and the original input video was calculated. This residual was coded using a fixed quantization scheme similar to the one used for coding of the base layer. The quantization parameter was fixed to one value for the whole sequence and used to quantize the DCT coefficients of nonoverlapping 8 x 8 blocks. The quantization parameter was varied ($Q = 2, 6, ..., 30$ and $38, 46, ..., 62$) to provide a wide range of quantization. The enhancement bandwidth was computed using the Huffman codes from Inter-prediction in the MPEG-2 standard.

3.4 Parameter Selection for Adaptive Deinterlacing

The algorithm used for parameter selection by the encoder depends on the particular frame partitioning scheme used. There is minimal processing for simulations using blocks of the same size. The deinterlacing modes for fixed block size experiments were selected by simply choosing the mode resulting in the best PSNR for each block. Since the block sizes are limited to three cases (16 x 16, 8 x 8 and 4 x 4), it is easy to see that fixed block size experiments have limited bitrate scalability and often will not meet a specific target enhancement layer bitrate. Parameter selection for adaptive block size partitioning is more complicated than fixed block size partitioning, but in addition to allowing bits to be concentrated where there are needed most, the use of variable sized deinterlacing blocks permits finer control of the enhancement layer bitrate.

Figure 3.14 shows the parameter selection algorithm used by Sunshine[8]. It begins by deinterlacing each 16 x 16 block with every deinterlacing mode and selecting the mode with the best performance, i.e. the lowest MSE. The enhancement bitrate is then computed and compared to the target bandwidth. If additional bandwidth is still available, blocks with the highest MSE are further subdivided and the best mode for each subblock was chosen for deinterlacing. The enhancement bitrate was recomputed and this procedure was repeated until the available enhancement bandwidth was exhausted. This provides a simple, straightforward method for the encoder to utilize all of the available bandwidth. However, it is suboptimal in the rate-distortion sense since it does not minimize the total MSE.

The encoder developed by Sunshine could make better, in fact optimal, decisions with the tradeoff being increased complexity in the encoder. This is because an optimal decision algorithm requires a comprehensive search over all the signal processing modes at every block size. Since the object of this thesis is to evaluate the potential of adaptive format conversion, the use of an optimal decision algorithm is more appropriate for this implementation. An optimal frame partitioning was computed in these simulations by using Lagrangian optimization with each coding unit representing a 16 x 16 block. Every possible permutation of each partitioning was examined to construct a rate-distortion curve for each 16 x 16 block where the number of modes required for the partitioning was used for the rate and MSE is the distortion measure. As shown in Table 3.1,

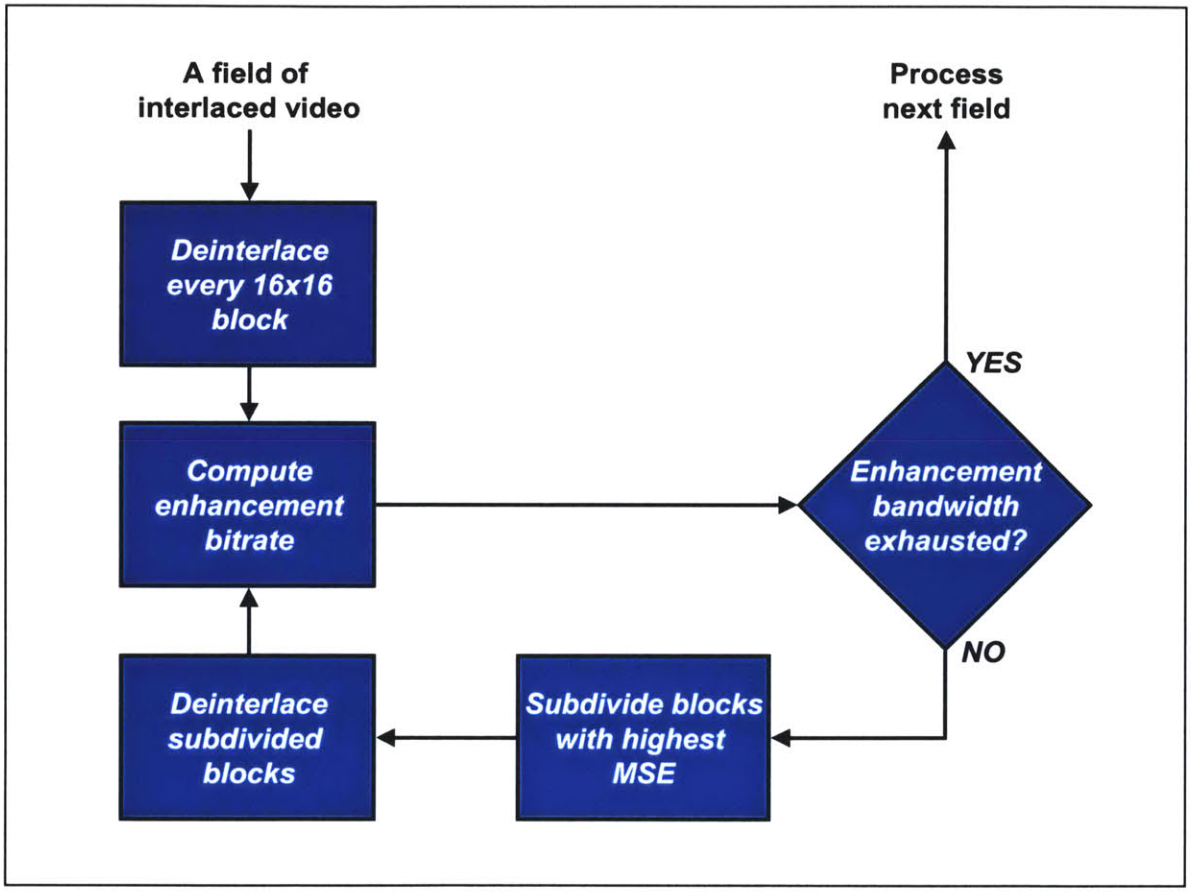

Figure 3.14: Recursive Parameter Selection. A straightforward method to utilize all of the available enhancement layer bandwidth. However, it does not provide optimal parameter selection.

there are six different partitionings and some of them have multiple permutations. Since permutations of the same partitioning have the same number of modes, only the permutations with the lowest distortion (for those partitionings that have more than one permutation) are needed to construct the rate-distortion curve. Figure 3.15 provides an example of the rate-distortion curve for a single 16 x 16 block. Lagrangian optimization was then performed using these rate-distortion curves to compute the optimal frame partitioning and corresponding deinterlacing modes for a given Lagrange multiplier. A bisection search can be used to find the proper Lagrange multiplier for a given enhancement bitrate.

A brief review of Lagrangian optimization is presented in the subsequent text. A detailed discussion of Lagrangian optimization can be found in [27, 28]. Define $R_{i,j}$ and $D_{i,j}$ to be the rate and the distortion of the ith block when the best jth partitioning is used. Let $x(i)$ denote any possible mapping for the partitioning of each block. We would like to solve the following budget constrained allocation problem: For a given total rate R_T, find the optimal mapping $x^\star(i)$ such that

$$\sum_i R_{i,x^\star(i)} \leq R_T \tag{3.7}$$

and a distortion metric $f(D_{1,x(1)}, D_{2,x(2)}, ..., D_{N,x(N)})$ is minimized.

The classical solution to this budget constrained allocation problem uses the discrete version of Lagrangian optimization. The basic idea of this technique is to introduce a Lagrange multiplier λ, which is a non-negative real number ($\lambda \geq 0$), and consider minimizing the Lagrangian cost function $J(\lambda) = \sum_i \left(D_{i,x(i)} + \lambda R_{i,x(i)}\right)$. Note that unlike the budget constrained allocation problem, there is no constraint on this minimization. Let $x^\star(i)$ be the mapping that minimizes $J(\lambda)$. This mapping $x^\star(i)$ is also the optimal solution to the budget constrained allocation problem formulated above when

$$R_T = \sum_i R_{i,x^\star(i)} \tag{3.8}$$

and minimizes the distortion metric $\sum_i D_{i,x(i)}$. Therefore, the minimization of the unconstrained Lagrangian cost function can be performed instead of the budget constrained allocation problem to obtain the desired solution. Since each node is coded independently,

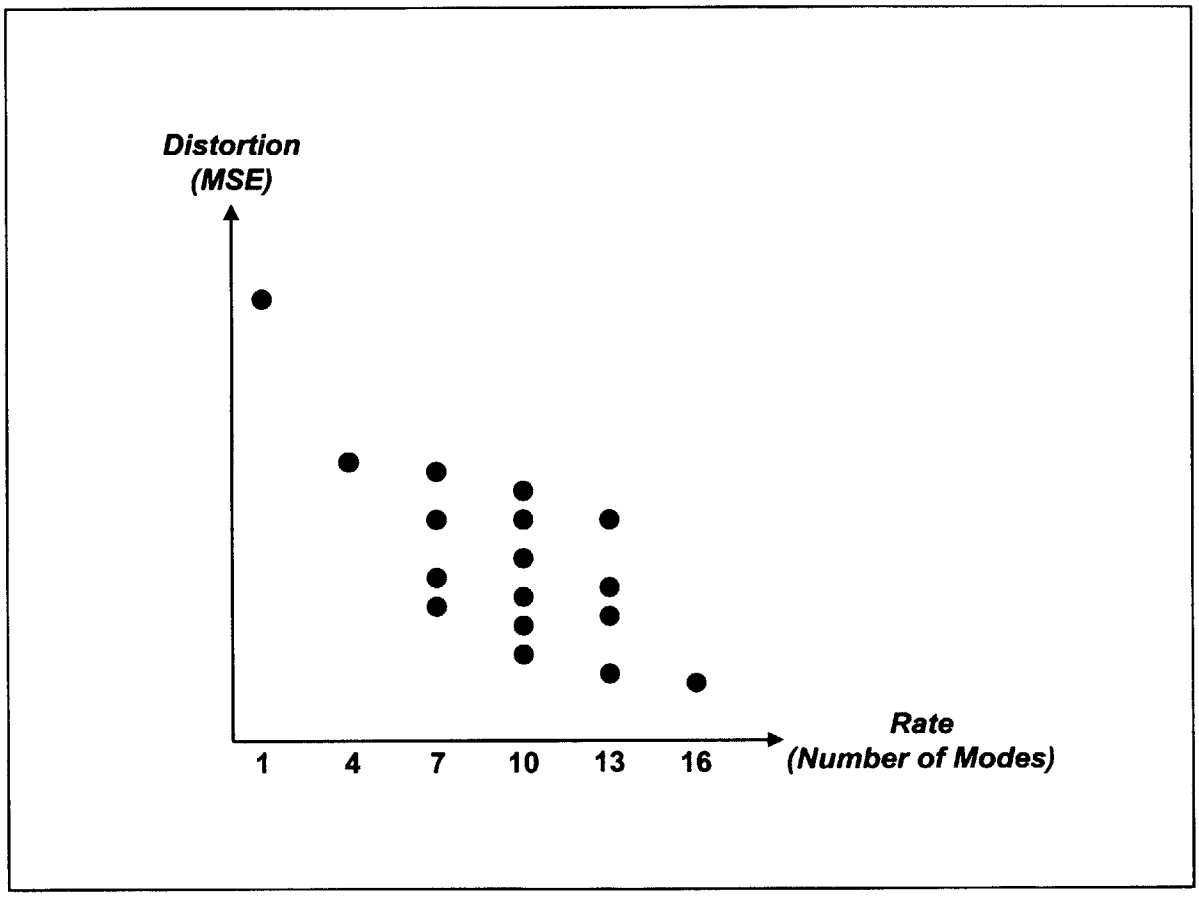

Figure 3.15: Rate-Distortion Curve. An example of the operating characteristics for the different partitionings of a single 16 x 16 block. Note that only the best permutation for each partitioning needs to be used for the optimization process.

the minimization can be computed independently for each block as follows:

$$\min J(\lambda) = \min \left[\sum_i \left(D_{i,x(i)} + \lambda R_{i,x(i)} \right) \right] \tag{3.9}$$

$$= \sum_i \left(\min \left[D_{i,x(i)} + \lambda R_{i,x(i)} \right] \right) \tag{3.10}$$

This analysis can be performed for various values of λ to create an operational rate-distortion curve. Note that when $\lambda = 0$, minimizing $J(\lambda)$ is equivalent to minimizing the distortion. Conversely, when λ becomes arbitrarily large, minimizing $J(\lambda)$ is equivalent to finding the minimum achievable rate. Intermediate values of λ can be used to determine intermediate operating points on the curve.

In the implementation overview, the desired distortion metric was the PSNR. Therefore, one may wonder why the MSE of each block was chosen to construct the rate-distortion curves for Lagrangian optimization instead of the PSNR of each block. The MSE is used as the distortion metric because Lagrangian optimization minimizes the sum of the individual distortions and minimizing the sum of the MSE of each block is equivalent to maximizing the total PSNR as shown below:

$$\arg\max PSNR_{total} = \arg\max \left[10 \log_{10} \left(\frac{255^2}{MSE_{total}} \right) \right] \tag{3.11}$$

$$= \arg\max \left[10 \log_{10} \left(\frac{255^2}{\sum_i MSE_i} \right) \right] \tag{3.12}$$

$$= \arg\max \left[10 \log_{10} 255^2 - 10 \log_{10} \sum_i MSE_i \right] \tag{3.13}$$

$$= \arg\min \left[\log_{10} \sum_i MSE_i \right] \tag{3.14}$$

$$= \arg\min \left[\sum_i MSE_i \right] \tag{3.15}$$

The relation between the distortion of each coding unit and the total distortion is not as clear for the PSNR metric as it is for the MSE metric. For example, the total PSNR is not equivalent to the sum (or even the average) of the PSNR of each block. For ease

of discussion, each of the N blocks are assumed to be the same size so that $MSE_{total} = \frac{1}{N}\sum_{i=1}^{N} MSE_i$.

$$PSNR_{total} = 10\log_{10}\left(\frac{255^2}{MSE_{total}}\right) \tag{3.16}$$

$$= 10\log_{10}\left(\frac{255^2}{\frac{1}{N}\sum_{i=1}^{N} MSE_i}\right) \tag{3.17}$$

$$= 10\log_{10} 255^2 - 10\log_{10}\left(\frac{1}{N}\sum_{i=1}^{N} MSE_i\right) \tag{3.18}$$

$$PSNR_{sum} = \sum_{i=1}^{N} PSNR_i \tag{3.19}$$

$$= \sum_{i=1}^{N} 10\log_{10}\left(\frac{255^2}{MSE_i}\right) \tag{3.20}$$

$$= 10N\log_{10} 255^2 - \sum_{i=1}^{N} 10\log_{10} MSE_i \tag{3.21}$$

$$PSNR_{average} = \frac{PSNR_{sum}}{N} \tag{3.22}$$

$$= 10\log_{10} 255^2 - \frac{1}{N}\sum_{i=1}^{N} 10\log_{10} MSE_i \tag{3.23}$$

Therefore,

$$PSNR_{total} \neq PSNR_{sum} \neq PSNR_{average} \tag{3.24}$$

3.5 Summary

This chapter began by examining the two basic types of information that can be coded in the enhancement layer of a scalable coding scheme. In addition to the well-known concept of residual coding, another type of information, adaptive format conversion, was

shown to be a promising type of enhancement information. The concept of adaptive format conversion has not been studied in great detail. An adaptive deinterlacing implementation by Sunshine provided some interesting results, however, those simulations suggest other issues that need to be examined to better understand the use of adaptive format conversion. Some of these issues include the lack of base layer coding, suboptimal mode selection and residual coding. These issues were addressed in a new implementation that will be used in various experiments described in the next chapter to examine the potential of adaptive format conversion to improve video scalability.

Chapter 4

Performance of Adaptive Format Conversion

The previous chapter described the implementation developed for this thesis to evaluate the potential of adaptive format conversion. This implementation includes two components not present in previous implementations: a base layer coder and a residual coder. The base layer coder will be used to examine the effect of base layer coding on the dependent enhancement layer. The residual coder will be used to both compare adaptive format conversion and residual coding as well as examine the use of both types of enhancement information. This chapter will begin by explicitly formulating the scalable coding problem to be examined. The use of the base layer distortion instead of the base layer rate will be important to separate the coding efficiencies of the base and enhancement layers. The scalable codec implementation is then used in different simulations to examine the effect of the base layer on adaptive format conversion, compare the two types of enhancement information and investigate the use of both types of enhancement information in the remainder of the chapter.

4.1 Problem Formulation

Consider a scalable coding scheme with two layers. Unlike simulcast coding where each layer is independent and the quality of each layer is controlled by only one variable (the rate of that layer), the dependency of the enhancement layer on the base layer complicates the analysis of scalable systems. Let R_b and D_b represent the rate and distortion of the base layer. Since the base layer is independently coded, the base layer distortion is dependent

only on the base layer rate, i.e.

$$D_b = f_1(R_b). \tag{4.1}$$

Let R_e and D_e represent the rate and distortion of the enhancement layer. The enhancement layer uses both the decoded base layer in addition to the enhancement layer bitstream, therefore the enhancement layer distortion is dependent on both the base layer and the enhancement layer rates, i.e.

$$D_e = f_2(R_b, R_e). \tag{4.2}$$

Another interpretation of the dependency of the enhancement layer can be obtained by inverting the base layer rate-distortion relation and substituting this into the previous equation to see that the enhancement layer distortion is a function of the base layer distortion and the enhancement layer rate, i.e.

$$D_e = f_2(R_b, R_e) \tag{4.3}$$
$$D_e = f_2(f_1^{-1}(D_b), R_e) \tag{4.4}$$
$$D_e = f_3(D_b, R_e) \tag{4.5}$$

The second viewpoint (the enhancement layer distortion is a function of the base layer distortion and the enhancement layer rate) will be used instead of the first viewpoint (the enhancement layer distortion is a function of the base layer rate and the enhancement layer rate). The difference between the viewpoints is subtle, but it is important because it separates the coding efficiency of the base and enhancement layer encoders. This separation is important because there are many different encoders that could be used for the base layer with a wide range of coding efficiencies and the focus of this thesis is on enhancement layer coding. Therefore, this thesis would like to avoid developing results that are applicable only to a specific base layer encoder. These results would have limited utility if one were not using the same encoder. By viewing the base layer in terms of its distortion, the results are more general. Separation of the base layer coding efficiency also simplifies the implementation and no attempt was made to optimize the coding efficiency of the base layer. This is not significant for the analysis in this thesis since the base layer bitrate is never used directly or in comparison to an enhancement layer bitrate.

The general scalable framework involves two independent variables (R_b and R_e) and two dependent variables (D_b and D_e). This framework is difficult to visualize and unwieldy for analysis. For example, it is unclear what is the tradeoff between the distortions of the two layers. Recall that the two layers have different formats, therefore it is difficult to quantitatively compare D_b and D_e. To make this problem more tractable, our approach will be to reduce the general problem to one with one independent variable and one dependent variable. This is accomplished by first coding the base layer and then examining the enhancement layer distortion as a function of the enhancement layer rate (for the particular base layer). Note that this analysis can be repeated for different base layers to sample the entire parameter space so there is no loss in generality. The redefined problem formulation is: Given a particular base layer, minimize the enhancement layer distortion (D_e) for a given enhancement layer rate (R_e).

4.2 Adaptive Format Conversion as Enhancement Information

The simplest manner to create video in the enhancement layer format from the base layer video is to perform nonadaptive format conversion for the entire sequence (with no residual coding). Note that this processing does not require transmission of an enhancement layer bitstream, i.e. $R_e = 0$. The resulting video often does not contain the high frequency detail of the original enhancement layer video since it is created solely from the coded base layer video and nonadaptive format conversion is limited in the detail it can recover. Despite the fact that this video may not contain much of the detail of the original enhancement layer, nonadaptive format conversion does result in video with the proper format and can be considered a default method to achieve this. Thus, it can be considered a reference point to compute the relative coding gains of other ways to construct the enhancement layer video. Since four different deinterlacing techniques are used in this thesis, we will choose the best case scenario for this reference point. Each of the four deinterlacing methods are applied nonadaptively over the whole sequence resulting in four different progressive sequences. The sequence with the highest PSNR will be used as this reference point and defined to be the Best NFC (Nonadaptive Format Conversion) point. Note that a Best NFC point is defined relative to the base layer and one can be defined for

every base layer.

Figure 4.1 shows the use of adaptive format conversion as enhancement information for the Carphone and News sequences. In these examples, the base layer was uncoded. Therefore, these results can be considered empirical upper bounds on the performance of adaptive format conversion since there are no artifacts from compression and the deinterlacing modes can exploit the perfect information in the base layer. The distortion of the enhancement layer (expressed in PSNR) is plotted as a function of the rate of the enhancement layer (expressed in BPP). Note the circles represent the best result from nonadaptive format conversion and these Best NFC points are plotted at 0 BPP Enhancement Layer. Unless otherwise noted, all PSNR gains from enhancement information in this thesis will be computed relative to the Best NFC point. The figures show that a significant improvement in the PSNR of the enhancement layer (gains from 4.38 dB to 6.1 dB for the Carphone sequence and gains from 8.51 dB to 10.1 dB for the News sequence) can be achieved by allocating a small amount of bandwidth (between 0.01 and 0.12 BPP) for adaptive format conversion information. Visual inspection showed a clear improvement in video quality with the use of adaptive format conversion. The bandwidth required to support adaptive format conversion ranges from less than 0.01 BPP to support the use of 16 x 16 blocks up to approximately 0.12 BPP to support the use of 4 x 4 blocks.

Figure 4.1 also demonstrates the differences between the two types of frame partitioning schemes for adaptive format conversion. The points connected by the solid red lines represent the three different fixed block size partitionings and the points connected by ther blue dotted lines represent the points achievable with adaptive block sizes. These simulations show many benefits to using adaptive block sizes. First, adaptive block sizes provide finer levels of bitrate scalability than the use of fixed block sizes. For example, fixed block size partitioning with 16 x 16 blocks and 8 x 8 blocks result in enhancement bitrates of 0.0075 BPP and 0.0305 BPP, respectively. It is possible that the available enhancement bandwidth is in between these two values and use of a fixed block size scheme would force the use of 16 x 16 blocks which would waste the remaining available bandwidth. One can easily see that there are many more operating points with an adaptive block size scheme. In fact, the points shown represent only a portion of those achievable and practically any target bitrate could be achieved with adaptive block sizes. The relatively small number of operating points is sufficient for discussion and was chosen to simplify the processing required by the simulations. The results with adaptive block sizes

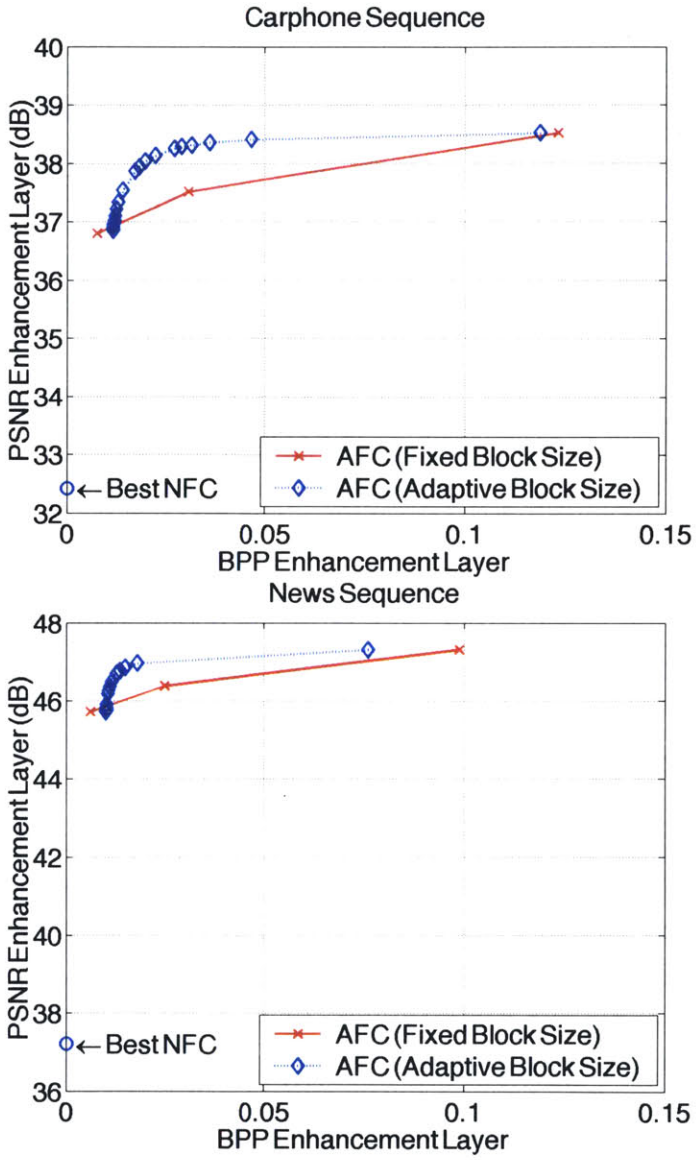

Figure 4.1: Adaptive Format Conversion as Enhancement Information for the Carphone (top) and News (bottom) Sequences. The base layer was uncoded in these simulations. PSNR gains in the enhancement layer from 4.38 dB to 6.1 dB can be achieved for the Carphone sequence by using adaptive format conversion (compared to the best result from nonadaptive format conversion). Similarly, PSNR gains in the enhancement layer from 8.51 dB to 10.1 dB can be achieved for the News sequence.

are generally more efficient (result in a higher PSNR) than fixed block sizes despite the additional overhead of the frame partitioning that needs to be transmitted in addition to the deinterlacing modes. The only scenario where fixed block size partitioning is more efficient is at the very low end of the achievable enhancement bitrates. This scenario occurs when the limited bandwidth does not permit much block division and the partitioning overhead is a significant fraction of the total enhancement layer bandwidth. Otherwise, adaptive block sizes allow the encoder to use available bits where they are needed resulting in higher coding efficiency. It is important to reiterate (as in the discussion of the implementation) the tradeoff for these benefits is the increased codec complexity required for adaptive block sizes.

Figure 4.2 provides another example of fixed block size partitioning and adaptive block size partitioning to provide more insight into the different partitioning schemes. The two pictures on the left are cropped sections of one frame of the Carphone sequence created from an uncoded base layer. Figure 4.2(a) was created using adaptive format conversion with nonoverlapping 8 x 8 blocks. Figure 4.2(b) illustrates the frame partitioning and mode selection for Figure 4.2(a). The four different deinterlacing modes are represented by different colors: linear interpolation (black), Martinez-Lim deinterlacing (dark gray), backward field repetition (light gray) and forward field repetition (white). Figure 4.2(c) was created using adaptive format conversion with adaptive block sizes. The enhancement layer bitrate for the adaptive block size partitioning (0.03 BPP) was the same as the bandwidth required for the fixed 8 x 8 block size partitioning. Figure 4.2(d) illustrates the frame partitioning and mode selection for Figure 4.2(c). Note how the use of adaptive block sizes allows the encoder to concentrate bits in areas of fine detail by using smaller blocks in those areas. Consistent with the results shown in Figure 4.1, the adaptive block size partitioning achieved a higher PSNR for the same enhancement bitrate in this example. (The resulting PSNR from the fixed block size and adaptive block size experiments was 37.52 dB and 38.32 dB, respectively.)

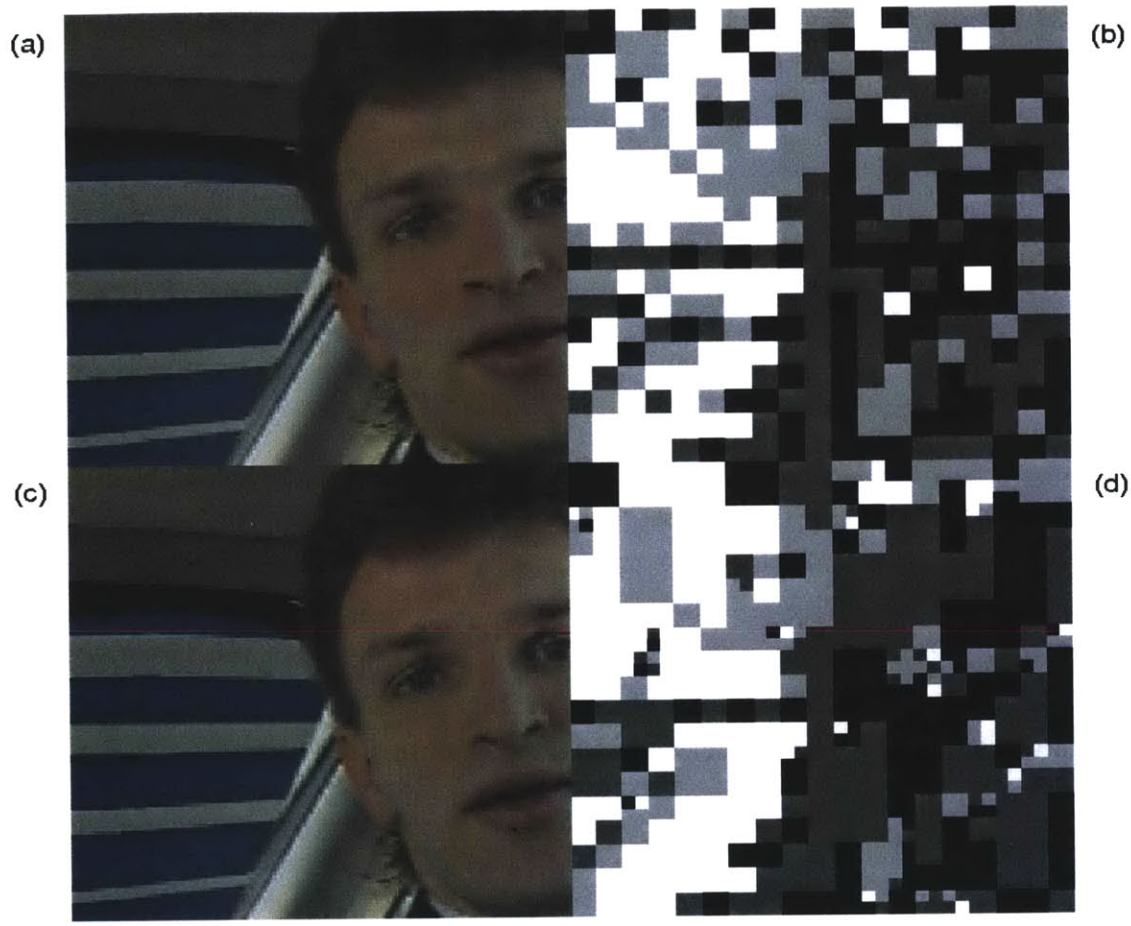

Figure 4.2: Frame Partitioning for Adaptive Format Conversion. Fixed and adaptive block sizes were examined for adaptive format conversion with the Carphone sequence. (a) A cropped section of one frame that was created using adaptive format conversion with fixed 8 x 8 blocks. (b) The frame partitioning and mode selection for the fixed 8 x 8 block size experiment. The modes are represented by different colors: linear interpolation (black), Martinez-Lim deinterlacing (dark gray), backward field repetition (light gray) and forward field repetition (white). (c) A cropped section of the same frame that was created using adaptive format conversion with adaptive block sizes. The enhancement bitrate is the same as that of the fixed 8 x 8 block size experiment. (d) The frame partitioning and mode selection for the adaptive block size experiment. Note how adapting the block size allows the encoder to concentrate bits in the areas of high detail.

4.3 Effect of Base Layer Coding on Adaptive Format Conversion

One of the limitations of the previous research on adaptive format conversion was that the simulations were performed with an uncoded base layer. Note that use of an uncoded base layer corresponds to zero distortion (an infinite PSNR for the base layer) since there is no loss of information. This is very significant because the deinterlacing modes have perfect information about the available fields in the base layer to reconstruct the missing fields in the enhancement layer. A compressed base layer would introduce compression artifacts that will hinder attempts to properly reconstruct the enhancement layer. Therefore, experimental results with an uncoded base layer can be considered upper empirical bounds on the performance of adaptive format conversion. It is important to note that the computation of upper performance bounds is very useful and may still be applicable when the base layer is coded robustly. However, it is unclear what the effect of the quality of the base layer will be on adaptive format conversion and simulations were performed to examine this issue.

Figure 4.3 provides an example of the use of adaptive format conversion as enhancement information for the Carphone and News sequences when the base layer is compressed. In these experiments, the quantizer in the base layer encoder was fixed at 10 for the entire sequence resulting in a base layer PSNR of 36.08 dB at 2.47 BPP for the Carphone sequence and a base layer PSNR of 35.62 dB at 2.94 BPP for the News sequence. The figures show that a significant improvement in the PSNR of the enhancement layer (gains from 2.72 dB to 3.78 dB for the Carphone sequence and gains from 2.18 dB to 2.87 dB for the News sequences) can be achieved by allocating a small amount of bandwidth for adaptive format conversion information. The PSNR gain is computed relative to the Best NFC point.

Note that the gains from adaptive format conversion in Figure 4.3 are much smaller than those seen in Figure 4.1 when the base layer is uncoded due to imperfect base layer information. The range of PSNR gains for the Carphone sequence have dropped from 4.38 dB to 6.1 dB with an uncoded base layer to 2.72 dB to 3.78 dB with the coded base layer. Similarly, the PSNR gains in the News sequence have dropped from 8.51 dB to 10.1 dB with an uncoded base layer to 2.18 dB to 2.87 dB with the coded base layer. This

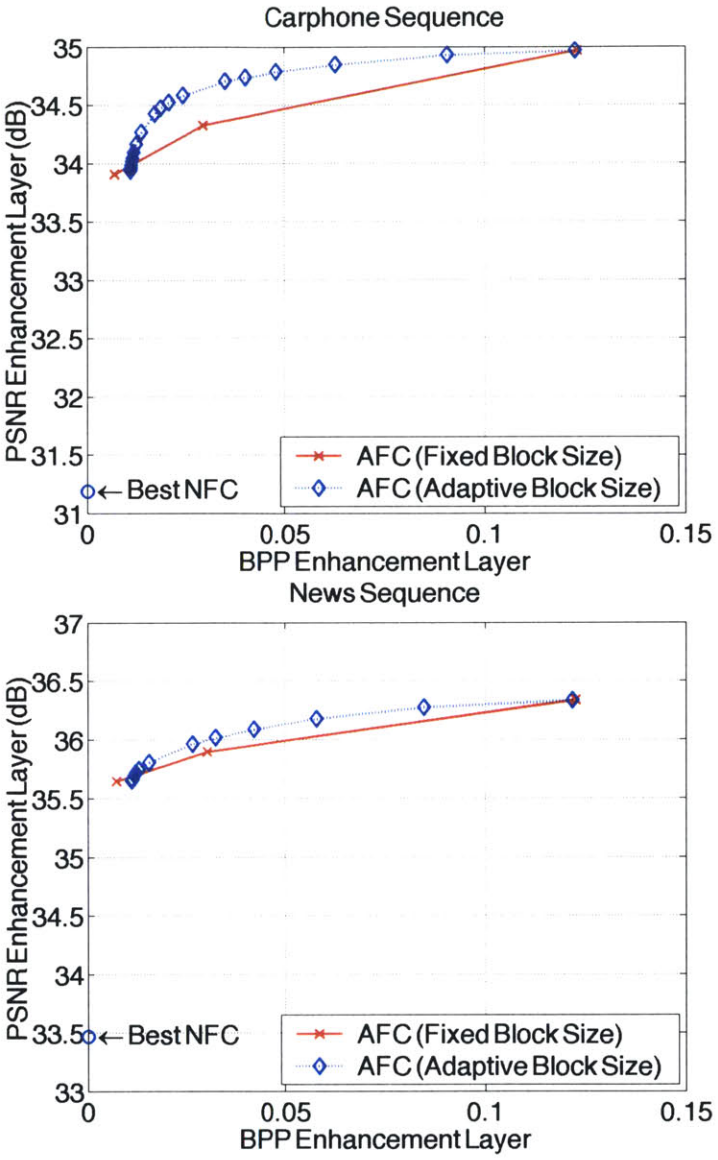

Figure 4.3: Adaptive Format Conversion as Enhancement Information for the Carphone (top) and News (bottom) Sequences. The base layer was coded with $Q = 10$ to provide an example of the effects of base layer compression on adaptive format conversion. PSNR gains in the enhancement layer from 2.72 dB to 3.78 dB can be achieved for the Carphone sequence by using adaptive format conversion (compared to the best result from non-adaptive format conversion). Similarly, PSNR gains in the enhancement layer from 2.18 dB to 2.87 dB can be achieved for the News sequence.

example suggests the need to further examine the effect of the base layer on adaptive format conversion. One method to do this is to investigate the coding gain from adaptive format conversion as a function of the base layer quality. This can be done by performing different adaptive format conversion simulations with different base layers.

Figure 4.4 illustrates the possible improvement to the Carphone and News sequences from adaptive format conversion as a function of the quality (distortion) of the base layer. The PSNR of the enhancement layer is plotted as a function of the PSNR of the base layer. Three curves are shown representing the achievable PSNR by using the best nonadaptive format conversion technique (red solid curve), adaptive format conversion with 16 x 16 blocks (green dashed curve) and adaptive format conversion with 4 x 4 blocks (blue dotted curve). The enhancement bitrate required to support the adaptive format conversion data in these simulations was comparable to the results shown earlier (approximately 0.01 BPP for 16 x 16 blocks and 0.12 BPP for 4 x 4 blocks). Adaptive format conversion with 16 x 16 and 4 x 4 blocks were selected to demonstrate the minimum and maximum possible gains from adaptive format conversion with this implementation since they are the coarsest and finest possible block partitionings. Note that operating points between the curves for adaptive format conversion with 4 x 4 blocks and 16 x 16 blocks can be achieved by using adaptive block sizes.

The labels #1 - #6 are used to describe the computation of the minimum and maximum possible gains for adaptive format conversion with 16 x 16 blocks and 4 x 4 blocks. The minimum gain for adaptive format conversion with 16 x 16 blocks is computed by subtracting Point #1 from Point #2 and results in gains of 0.95 dB and 0.71 dB for the Carphone and News sequences, respectively. The maximum gain for adaptive format conversion with 16 x 16 blocks is computed by subtracting Point #4 from Point #5 and results in gains of 4.17 dB and 6.78 dB for the Carphone and News sequences, respectively. The minimum gain for adaptive format conversion with 4 x 4 blocks is computed by subtracting Point #1 from Point #3 and results in gains of 1.66 dB and 1.28 dB for the Carphone and News sequences, respectively. The maximum gain for adaptive format conversion with 4 x 4 blocks is computed by subtracting Point #4 from Point #6 and results in gains of 5.81 dB and 7.97 dB for the Carphone and News sequences, respectively. These gains are summarized in Table 4.1.

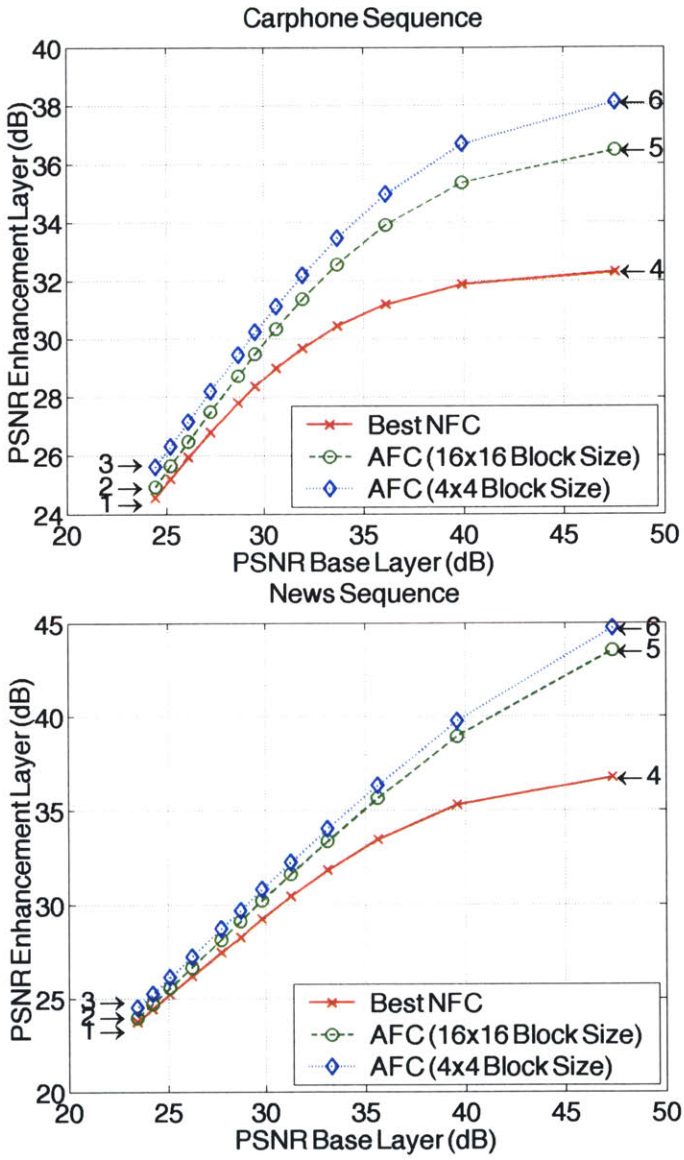

Figure 4.4: Effect of Base Layer on Adaptive Format Conversion for the Carphone (top) and News (bottom) Sequences. Large gains can be achieved when the base layer is coded well (up to 5.81 dB and 7.97 dB improvement for the Carphone and News sequences, respectively) and PSNR gains of ~1 dB were present even when the base layer was coded poorly. See text for explanation of points #1 - #6 used for computing gains.

	Carphone Sequence	**News Sequence**
Min. Gain 16 x 16 blocks	0.95 dB	0.71 dB
Max. Gain 16 x 16 blocks	4.17 dB	6.78 dB
Min. Gain 4 x 4 blocks	1.66 dB	1.28 dB
Max. Gain 4 x 4 blocks	5.81 dB	7.97 dB

Table 4.1: Minimum and Maximum Possible Gains for Adaptive Format Conversion With 16 x 16 Blocks and 4 x 4 Blocks. The minimum gain is computed by examining the coarsest coded base layer ($Q = 30$). The maximum gain is computed by examining the finest coded base layer ($Q = 2$).

These results not only show that large gains can be achieved when the base layer is coded well (up to 5.81 dB and 7.97 dB improvement for the Carphone and News sequences, respectively), but that adaptive format conversion can provide substantial improvement in the enhancement layer even when the base layer is coded coarsely (over 1 dB gains can be achieved with adaptive format conversion in both sequences even when the base layer was coded with $Q = 30$).

4.4 Comparison of Adaptive Format Conversion and Residual Coding

One of the key differences between adaptive format conversion and residual coding is the different rates that each data type can achieve. A smaller number of parameters is usually needed for coding a region using adaptive format conversion compared to residual coding. This is due to the fact that the format conversion modes are transmitted in adaptive format conversion compared to a group of coefficients in residual coding. For example, a maximum of 16 modes would need to be transmitted for each 16 x 16 block in adaptive format conversion with the implementation described in this paper (since the smallest block size is 4 x 4) while up to 256 coefficients per 16 x 16 block may need to be transmitted for residual coding. In addition, the bits per mode in adaptive format conversion is usually substantially less that the bits per coefficient in residual coding. This enables adaptive format conversion to provide video scalability at low enhancement bitrates that are often not possible with residual coding, even with the coarsest residual quantizer.

Adaptive format conversion and residual coding also have different distortions that they can achieve. The different format conversion methods used in adaptive format conversion are limited in the detail that they can recover since they are dependent on the decoded base layer. Residual coding does not have this limitation since the prediction error is coded, thus residual coding can recover (practically) all of the video detail, albeit this may require use of a very fine quantizer which will result in an extremely high enhancement bitrate.

The different achievable rates and distortions between adaptive format conversion and residual coding suggest that a scalable coding scheme using only adaptive format

conversion with no residual coding should be compared to another scheme which utilizes a fixed method of format conversion for the entire sequence followed by residual coding. The analysis in this section is performed to provide insight into the differences between the two types of enhancement data. Note that adaptive format conversion and residual coding do not have to be used exclusively and the use of both types of enhancement information will be examined in the next section.

Figure 4.5 compares adaptive format conversion and residual coding for the Carphone and News sequences. In these experiments, the base layer was uncoded to provide empirical upper bounds on the gain from each type of enhancement data. The circles in the figure represent the highest PSNR achieved of the four sequences created using each of the deinterlacing modes on the whole sequence, i.e., nonadaptive format conversion. The results illustrate the ability of adaptive format conversion to provide video scalability at low enhancement bitrates (between 0.01 and 0.05 BPP) that are not possible even with the coarsest quantizer for residual coding. Note that the improvement in the enhancement layer quality from adaptive format conversion is quite substantial. Even at the lowest achievable bitrate (which corresponds to adaptive format conversion with 16 x 16 blocks), there is a 4.4 dB gain and a 8.5 dB gain for the Carphone and News sequences, respectively. Visual inspection showed a substantial improvement in video quality. The curve for adaptive format conversion has an exponential shape demonstrating the limitation in the detail of the original input video that it can recover. This limitation may be due to the limited resolution (4 x 4 blocks) of the adaptive format conversion implementation used in this paper. However, the exponential shape of the curve suggests that allowing smaller block sizes (e.g. 2 x 2) will probably not improve the performance. Even though the base layer is uncoded, it is important to note that it does not contain the same information as the original enhancement layer and therefore adaptive format conversion will be limited in the detail it can recover. Despite the fact that the adaptive format conversion curve quickly tapers off above $\sim$0.02 BPP, it is still more efficient than residual coding over the common range of bitrates that both types of enhancement data can achieve.

Figure 4.6 also compares adaptive format conversion and residual coding for the Carphone and News sequences. The difference between this figure and Figure 4.5 is that the base layer was coded with fixed quantization ($Q = 30$) for the whole sequence. This provides an example of the effects of a compressed base layer on scalable coding.

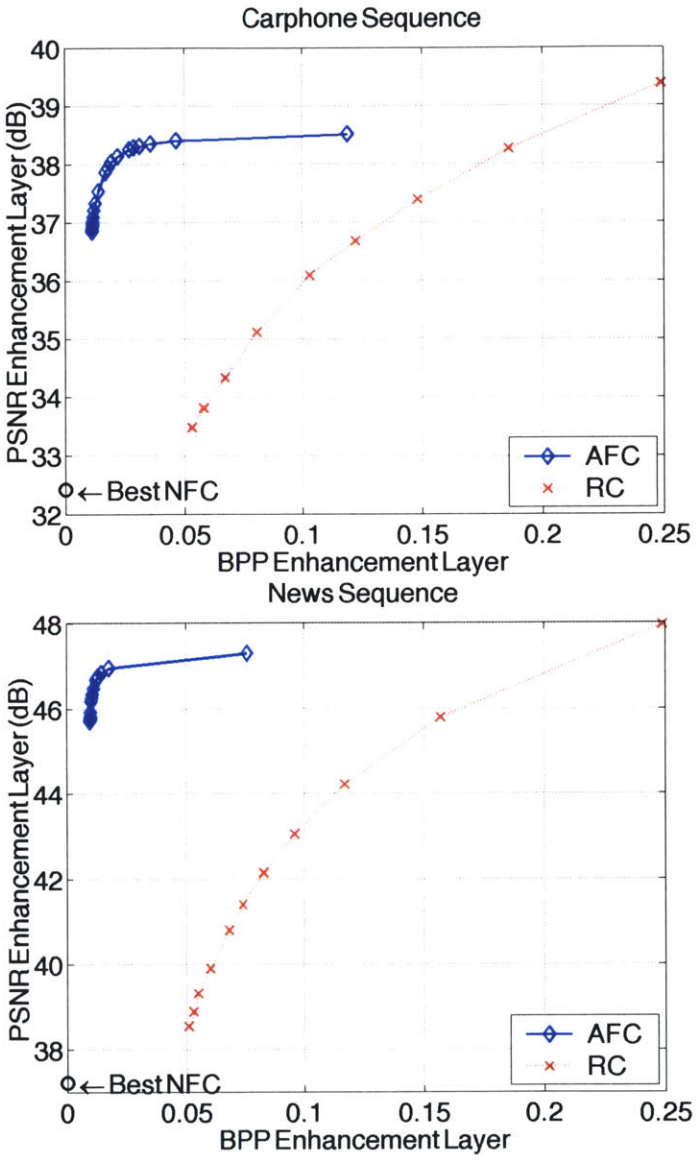

Figure 4.5: Comparison of Adaptive Format Conversion (AFC) and Residual Coding (RC) for the Carphone (top) and News (bottom) Sequences. The base layer was uncoded in these figures to determine empirical upper bounds on the performance of adaptive format conversion and residual coding. The circles represent the highest PSNR achieved using nonadaptive format conversion (NFC) with each of the four deinterlacing modes.

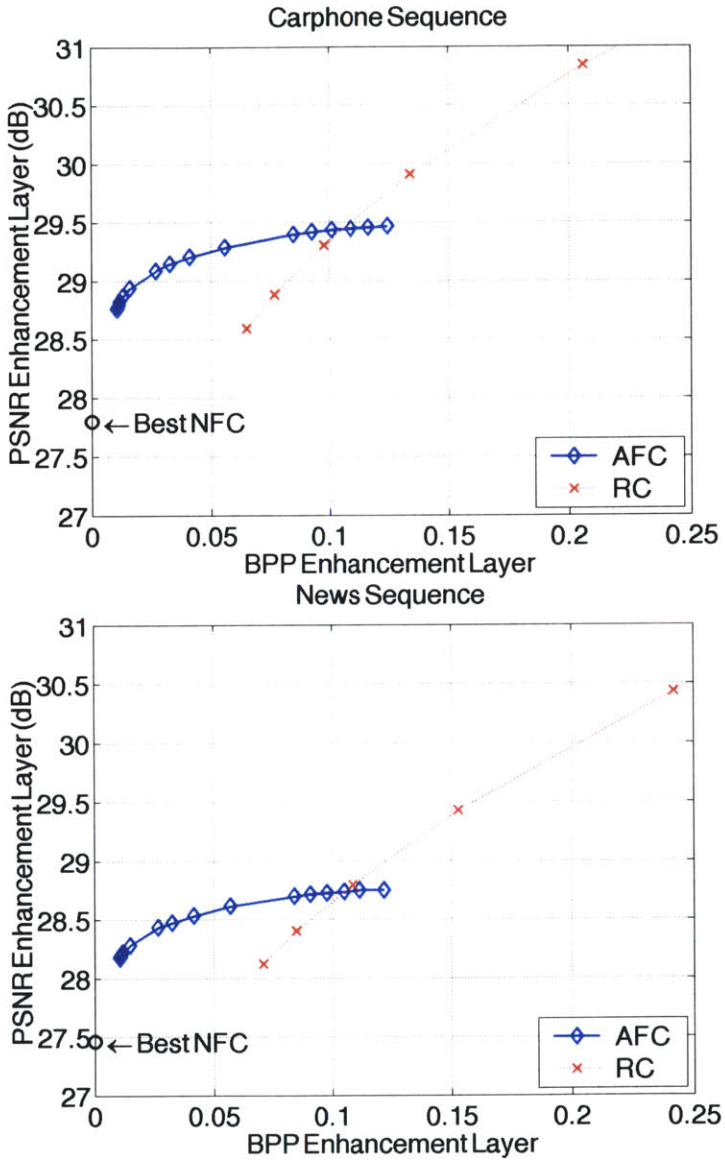

Figure 4.6: Comparison of Adaptive Format Conversion (AFC) and Residual Coding (RC) for the Carphone (top) and News (bottom) Sequences. The base layer was coded with fixed quantization ($Q = 30$) resulting in a base layer PSNR of 28.65 dB and 27.69 dB for the Carphone and News sequences, respectively. The circles represents the highest PSNR achieved using nonadaptive format conversion (NFC) with each of the four deinterlacing modes.

Comparison of Figures 4.5 and 4.6 show different relationships between adaptive format conversion and residual coding. The figures show that inexact base layer information affects adaptive format conversion significantly more than residual coding. The range of PSNR gains from adaptive format conversion have dropped from 4.4 dB to 6.1 dB with an uncoded base layer to 1.0 dB to 1.7 dB with the coded base layer for the Carphone sequence. Similarly, the gains in the News sequence have dropped from 8.5 dB to 10.1 dB with an uncoded base layer to 0.7 dB to 1.3 dB with the coded base layer. A similar dropoff in PSNR gain is not present with residual coding. This effect is due in part to the high dependence of adaptive format conversion on accurate base layer information. This demonstrates how adaptive format conversion is more susceptible to the propagation of quantization error than residual coding which uses the error difference. The propagation of quantization error causes residual coding to be more efficient than adaptive format conversion in part of their common bandwidth ranges (0.1 - 0.125 BPP for both sequences). Note that this result was not seen in Figure 4.5 where the simulations were performed with an uncoded base layer.

Figures 4.5 and 4.6 illustrate that five distinct coding situations can occur for a given base layer quality and enhancement bitrate:

- No video scalability is possible with either adaptive format conversion or residual coding because the enhancement bitrate is too low.
- Only adaptive format conversion can be used.
- Both types of enhancement data can be used, but adaptive format conversion is more desirable (i.e., results in a higher PSNR) than residual coding.
- Both types of enhancement data can be used, but residual coding is more desirable than adaptive format conversion. (Note that this situation was not seen in the uncoded base layer example.)
- Only residual coding can be used.

The different coding situations are illustrated in Figure 4.7 where a wide range of base layer qualities and enhancement bitrates are examined. The possible types of enhancement data for a given enhancement bitrate are plotted as a function of the base layer

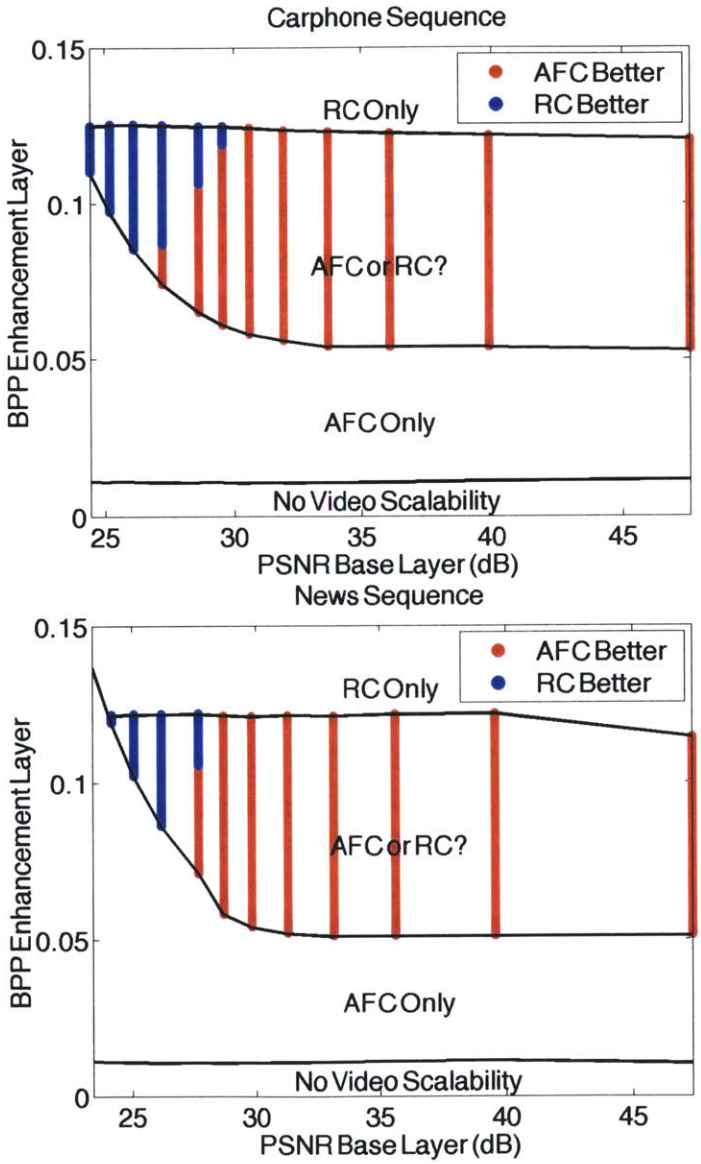

Figure 4.7: Comparison of Adaptive Format Conversion (AFC) and Residual Coding (RC) for the Carphone (top) and News (bottom) Sequences. Regions indicating the possible types of enhancement data for a given enhancement bitrate are shown as a function of the base layer distortion.

distortion (PSNR). The figure shows that video scalability is not possible with either type of enhancement data below 0.01 BPP and defines regions where adaptive format conversion and/or residual coding can be used. In the region where both types of enhancement data can be chosen, the red points represent situations where the use of adaptive format conversion is preferable (i.e., will result in a higher PSNR) and blue points represent situations where the use of residual coding is preferable. One may notice that the boundaries for adaptive format conversion are independent of the base layer distortion while the lowest enhancement bitrate where residual coding is possible starts to sharply increase when the base layer PSNR is low. This is not surprising since the enhancement bitrate for adaptive format conversion is directly related to the number of blocks that are coded and this is not dependent on the base layer quality. On the other hand, the number of coefficients that are coded in residual coding is dependent on the prediction error which is a function of the base layer quality. Two major results about adaptive format conversion can be seen from the figure. First, adaptive format conversion is the only method to provide video scalability at "low" enhancement bitrates (between 0.01 BPP and 0.05 BPP) regardless of the base layer quality. Second, adaptive format conversion is often superior to residual coding at providing video scalability at "medium" enhancement bitrates (between 0.05 BPP and 0.12 BPP). Residual coding starts to outperform adaptive format conversion when the base layer is not coded very well (when the base layer PSNR drops below ~28 dB). It should be noted that a scalable coding application with a low quality base layer may defeat the purpose of scalability since reception of a poor base layer may not be useful.

4.5 Use of Both Adaptive Format Conversion and Residual Coding

Adaptive format conversion and residual coding can be conceptualized as different types of enhancement data, but one does not need to use them exclusively and can incorporate both data types in a scalable scheme if desired. The last section showed that adaptive format conversion can provide video scalability at low enhancement bitrates, but it is useful to investigate whether adaptive format conversion can also improve coding efficiency at higher bitrates. This can be achieved by using both types of enhancement data and com-

paring it to the use of nonadaptive format conversion with residual coding. One scenario where this analysis would be applicable is the decision of whether or not to add adaptive format conversion to an existing scalable coding system that utilizes only residual coding for enhancement data.

Figure 4.8 examines the use of both adaptive format conversion and residual coding for the Carphone and News sequences. In this figure, the base layer was uncoded to provide an empirical upper bound on the possible gain. For simplicity, only two different types of adaptive format conversion will be combined with residual coding in this section: adaptive format conversion with 16 x 16 blocks (blue curve) and 4 x 4 blocks (magenta curve). These two examples represent the smallest and largest possible PSNR gains, respectively, from adaptive format conversion in the implementation described in this paper. The red curve represents the use of residual coding only (with nonadaptive format conversion). Note that a wide range of PSNR gains between these two special cases can be achieved with adaptive block sizes. The results show that inclusion of adaptive format conversion to a residual coder improves the coding efficiency at both "low" and "high" enhancement bitrates. An example of improved coding efficiency at a "low" enhancement bitrate is seen in the Carphone sequence where a PSNR of 37 dB can be achieved with 0.05 BPP when using both types of enhancement data (adaptive format conversion with 16 x 16 blocks and residual coding) compared to 0.13 BPP when using only residual coding. Thus, the use of adaptive format conversion results in a 62% reduction in the enhancement layer bandwidth. An example of improved coding efficiency at a "high" enhancement bitrate is also seen in the Carphone sequence where a PSNR of 43 dB can be achieved with 0.3 BPP when using both types of enhancement data (adaptive format conversion with 16 x 16 blocks and residual coding) compared to 0.6 BPP when using only residual coding. Thus, the use of adaptive format conversion provides a 50% reduction in the enhancement layer bandwidth.

Figure 4.9 also examines the use of both adaptive format conversion and residual coding for the Carphone and News sequences. The difference between this figure and Figure 4.8 is that the base layer was coded with fixed quantization ($Q = 30$) for the whole sequence to demonstrate the effect of a compressed base layer. Note that the results of Figure 4.9 are very different from those in Figure 4.8. In this figure, the three curves are very similar, thus adaptive format conversion does not provide significant compression

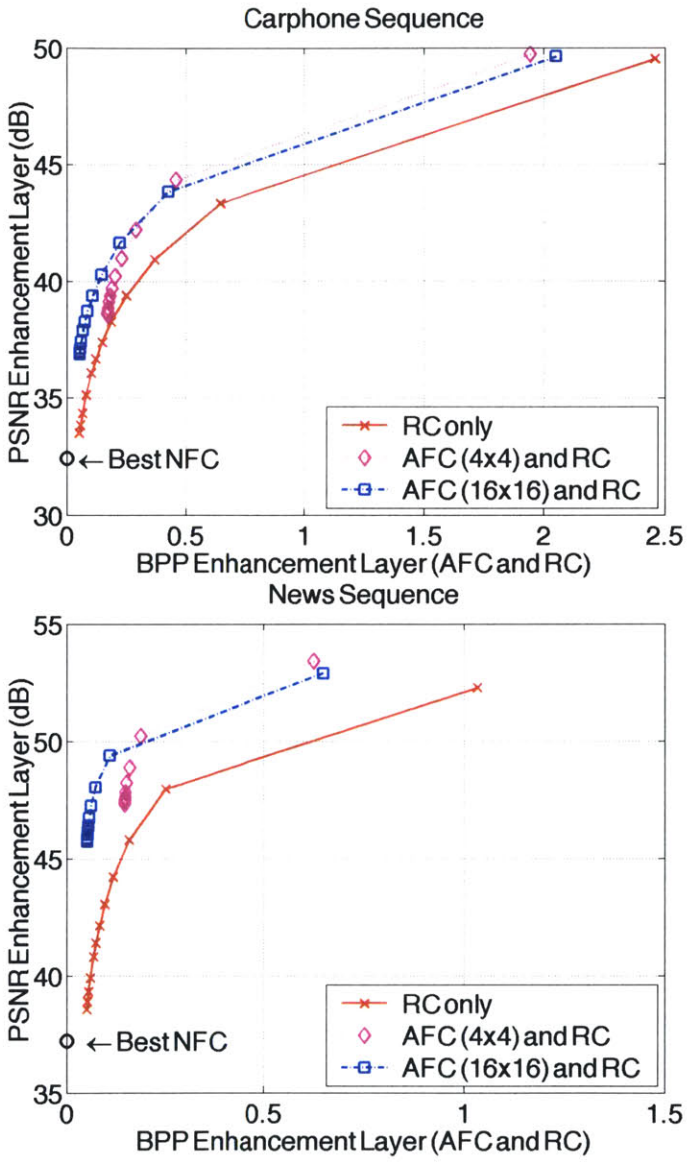

Figure 4.8: The Use of Both Adaptive Format Conversion (AFC) and Residual Coding (RC) for the Carphone (top) and News (bottom) Sequences. The base layer was uncoded to determine empirical upper bounds on the performance of adaptive format conversion and residual coding. The coding efficiency is improved with the inclusion of adaptive format conversion in this example.

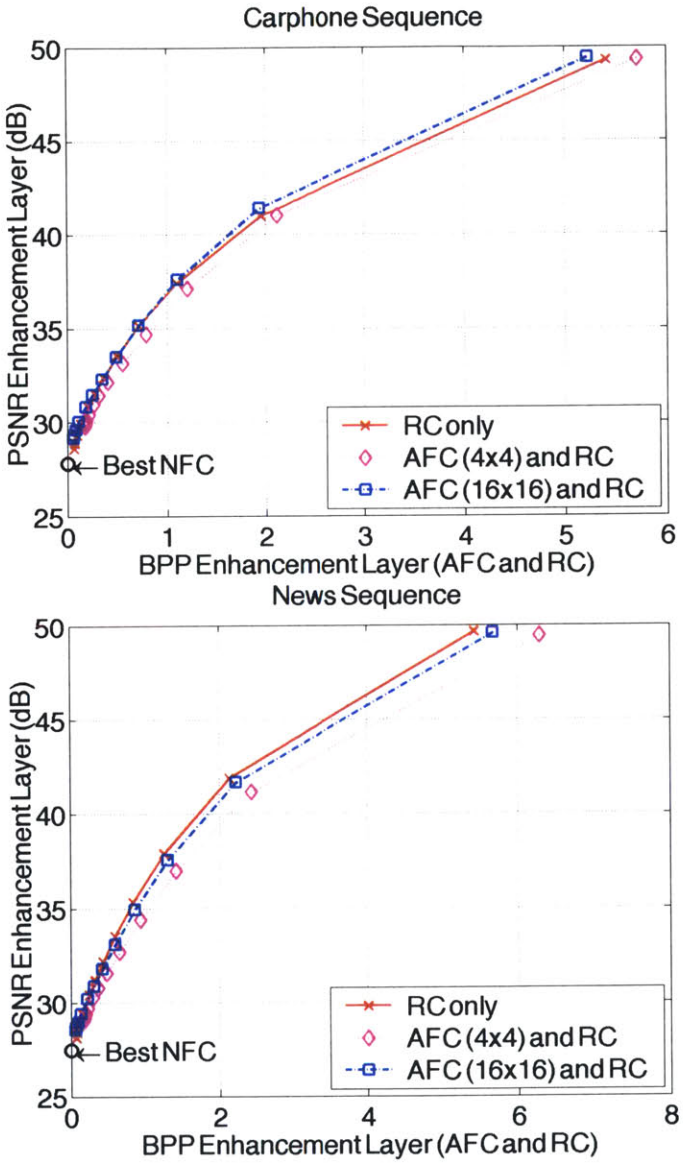

Figure 4.9: The Use of Both Adaptive Format Conversion (AFC) and Residual Coding (RC) for the Carphone (top) and News (bottom) Sequences. The base layer was coded with fixed quantization ($Q = 30$) resulting in a base layer PSNR of 28.65 dB and 27.69 dB for the Carphone and News sequences, respectively. The curves are very similar indicating no benefit to addition of adaptive format conversion to a residual coder when the base layer is not coded well.

gains with this particular base layer. In fact, adaptive format conversion with 4 x 4 blocks and residual coding provides worse compression than using only residual coding for both sequences. The different results seen in Figures 4.8 and 4.9 illustrate the high dependence of adaptive format conversion on the quality of the base layer.

The effect of the base layer distortion on the use of both enhancement data types is further examined in Figure 4.10. The results are plotted using the PSNR Gain which is defined to be the difference in PSNR between using adaptive format conversion with 16 x 16 blocks followed by residual coding and using only residual coding. This PSNR Gain is plotted as a function of the enhancement layer bitrate. Each curve represents a different base layer. Results for an uncoded base layer (which corresponds to Figure 4.8) are shown along with base layers that were coded with different quantization parameters (Q) that were fixed over the whole sequence (the curve with $Q = 30$ corresponds to Figure 4.9). The figure clearly shows the dependence of adaptive format conversion on the base layer distortion with better performance resulting from better base layer information. Note that the use of both enhancement data types is preferable (the curve takes on positive values) to using only residual coding for almost all the examples shown. The only curves where adaptive format conversion has a negative effect is when the base layer is coded rather poorly ($Q = 30$ and $Q = 62$). Thus, adaptive format conversion can improve video scalability when combined with residual coding at higher enhancement bitrates as long as the base layer is not coded poorly.

4.6 Summary

This chapter began by explicitly defining the scalable coding formulation to be examined. Different aspects of adaptive format conversion were then analyzed beginning with the effect of the base layer coding on adaptive format conversion. Adaptive format conversion proved to be useful over a wide range of base layer qualities. Large gains were achieved when the base layer was coded well and PSNR gains of over 1 dB were present even when the base layer was coded poorly. The comparison of adaptive format conversion and residual coding not only explicitly showed that adaptive format conversion can provide video scalability at low bitrates not possible with residual coding, it also showed that the use of adaptive format conversion instead of residual coding is generally prefer-

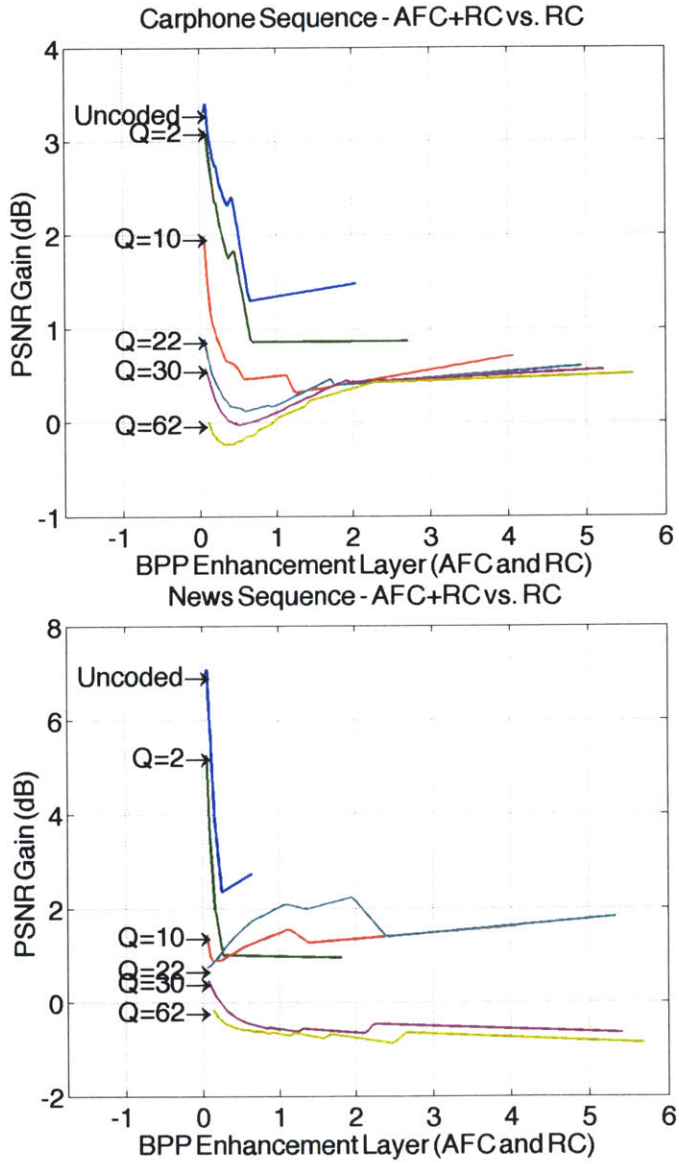

Figure 4.10: The Use of Both Adaptive Format Conversion (AFC) and Residual Coding (RC) for the Carphone and News Sequences. Each curve represents a different base layer. Results with an uncoded base layer (Uncoded) are shown along with base layers coded with the quantization parameter fixed to different levels ($Q = 2, 10, 22, 30, 62$).

able when the use of either type of enhancement data is possible. Experiments using both types of enhancement data showed that adaptive format conversion could assist residual coding and provide better coding efficiency than residual coding alone. These results support the utility of adaptive format conversion and will be applied in the next chapter to an important application, the migration path of digital television.

Chapter 5

Migration Path for Digital Television

The previous chapter examined various aspects of adaptive format conversion and showed how it can be used as an efficient type of enhancement information for scalable coding. These results will be applied in this chapter to an important application: the migration path for digital television. This chapter will begin by reviewing the U.S. digital television standard. The recently established standard has many improvements over the current analog NTSC standard, but there are still limitations on the transmittable video formats. The need to transmit higher resolution formats in the future has already been recognized and should be done in a backward-compatible manner so as not to render current digital television equipment obsolete. This problem is of great interest to the video processing community and is referred to as the migration path problem. The discussion in this chapter will show that the use of adaptive format conversion in a scalable coding scheme is ideally suited to the migration path.

5.1 U.S. Digital Television Standard

5.1.1 Overview

The development of digital television for terrestrial broadcasting in the United States began in September 1987 when the Federal Communications Commission (FCC) chartered the Advisory Committee on Advanced Television Service (ACATS). This advisory committee was developed to jump-start competitive testing and initiate the standardization process with the eventual goal of recommending an advanced television standard to the FCC to replace the analog NTSC standard that has been the national terrestrial broadcasting standard since 1953. ACATS asked industries, universities and research laboratories to propose advanced television systems in 1988. While ACATS was reviewing the many

different proposals that were submitted, the FCC made a significant decision in March 1990 to use a simulcast approach for the new advanced television system rather than a receiver-compatible approach. A receiver-compatible system would allow current NTSC television sets to generate a viewable picture from the new advanced television signal. This approach was used when color was introduced to the NTSC signal to allow existing black-and-white sets to not be obsolete. Receiver-compatibility was not feasible for a new television system since the display formats considered contained such a large amount of information that inclusion of the inefficient NTSC signal for backward-compatibility would make the new system highly inefficient. A simulcast broadcasting approach meant that the new television signal would have to be transmitted separately from NTSC broadcasts and a NTSC television would not be able to generate a picture from the advanced television signal.

Years of development and testing resulted in four different all-digital systems reaching a stalemate in February 1993. ACATS could not recommend one system in particular since each of the systems excelled in different aspects. Following a suggestion by ACATS, the companies decided to work together and formed the Grand Alliance in May 1993 to develop a single system that would attempt to combine the best features from the individual systems and a standard based on this prototype system would be recommended to the FCC for standardization. The members of the Grand Alliance were AT&T, General Instrument Corporation, Massachusetts Institute of Technology, Philips Electronics North America Corporation, David Sarnoff Research Center, Thomson Consumer Electronics and Zenith Electronics Corporation. The best technical elements of each system were combined, further improvements were made and ACATS recommended a standard based on the Grand Alliance prototype to the FCC in November 1995. This standard restricted broadcasters to 18 video formats. The FCC adopted this recommendation in December 1996 with the exception of the restriction on the proposed transmission formats. The FCC decided to remove the restriction on the possible transmission formats to allow market forces to decide the best formats to use. In the fall of 1998, commercial broadcast of digital television started in the United States.

The U.S. digital television standard [29, 30] incorporates many technological advances made over the past few decades. As a result, digital television systems are significantly better than their analog counterparts which are based on the NTSC standard that was developed in the 1940's and 1950's. In addition to delivering spectacular video and

multi-channel, compact disc quality sound, the new digital systems have features absent in conventional analog systems such as auxiliary data channels and easy interoperability with computers. [31, 32, 33, 34]

In addition to video coding, the digital standard also specifies the transmission of audio and auxiliary data. The focus of this thesis is on video processing, so the discussion will be restricted to the video portion of the standard. The video coder is based on the Main Profile implemented at High Level (MP@HL) within the MPEG-2 standard. [2, 3, 14, 35] This video coder has the following characteristics:

- Ability to handle both progressive and interlaced scanned video
- Upper bounds of 1920 $\frac{samples}{line}$, 1152 $\frac{lines}{frame}$ and 60 $\frac{frames}{second}$
- Maximum bit rate of 80 Mbits per second (Mbps)
- Maximum sample rate [1] of 62.6 Msamples per second

Video compression is achieved by using the discrete cosine transform (DCT) to exploit spatial redundancy and block-based motion estimation/compensation to exploit temporal redundancy in the video sequence. The energy compaction properties of the DCT are widely known [16] and fast, low-cost implementations have been developed, making DCT-based compression algorithms very popular. Motion estimation and compensation is performed on a block-by-block basis in order to predict a block in the current frame from adjacent frames.

5.1.2 Transmission Formats

Two of the most apparent differences between the new digital systems and the conventional analog systems are in the picture resolution and aspect ratio (the ratio between the number of pixels per line and the number of lines of resolution). The digital system is commonly referred to as *high-definition television* (HDTV) because of the increased picture resolution. Unlike the analog NTSC standard, which has one resolution format (480

[1] The sample rate is defined to be the pixel rate of the uncompressed video sequence.

lines with interlaced scanning at 59.94 fields/sec) with a 4:3 aspect ratio, the digital television standard allows any format as long as the upper bounds on the samples per line, lines per frame, frames per second, maximum bit rate and maximum sample rate are not exceeded. The 16:9 aspect ratio has been found to be more aesthetically pleasing, thus most of the television community has agreed that this aspect ratio will be used for HDTV broadcasting. Therefore, this thesis will assume that all HDTV formats have a 16:9 aspect ratio. One reason that the digital television standard allows multiple video formats is to permit source-dependent coding. For example, the broadcast of sporting events should be performed at the highest possible temporal rate (60 frames/sec with progressive scanning) to preserve the fast motion of the video. On the other hand, video generated from film suggests the use of a high spatial resolution to preserve the detail of the film and a low temporal resolution since film is typically recorded at lower frame rates such as 24 frames/sec. Examples of video formats are shown in Table 5.1. Note that all of the video formats in Table 5.1 are permitted in the digital standard except for the last two formats: 1080P and 1440P, which have sample rates that exceed the MPEG-2 MP@HL specification.

5.2 Migration Path

5.2.1 Limitations of the Digital Television Standard

Most of the television processing community has been focused on the current transition from the analog NTSC standard to the digital television standard. Many resources are being used to develop devices to assist this transition such as set-top boxes that convert a digital signal for display on analog televisions for consumers who want to delay purchasing a digital television but still want to be able to view digital content. The transition to digital television will take some time, but eventually, digital televisions will become the norm instead of the exception as they are today. To speed up this transition, the FCC stated when the standard was first adopted in 1996 that it would reclaim the broadcast spectrum of analog terrestrial channels in 2006. This deadline would allow only digital broadcasts over terrestrial channels and encourage both broadcasters and consumers to make the transition to digital television. While it is currently unclear whether this deadline will be adhered to or should be delayed since there is currently a debate as to whether

Format	Spatial Resolution	Scan Mode	Frame/Field Rate	Sample Rate (Msamples/sec)
720P@24fps	720 x 1280	PS	24 frames/sec	22
720P@30fps	720 x 1280	PS	30 frames/sec	28
720P	720 x 1280	PS	60 frames/sec	55
1080I	1080 x 1920	IS	60 fields/sec	62
1080P@24fps	1080 x 1920	PS	24 frames/sec	50
1080P@30fps	1080 x 1920	PS	30 frames/sec	62
1080P	**1080 x 1920**	**PS**	**60 frames/sec**	**124**
1440P	**1440 x 2560**	**PS**	**60 frames/sec**	**221**

Table 5.1: Examples of Video Formats. A spatial resolution of C x D represents C lines of vertical resolution with D pixels of horizontal resolution. Note that all of these video formats have a 16:9 aspect ratio. The scan mode is either progressive scan (PS) or interlaced scan (IS). The frame/field rate refers to the number of frames/sec for progressive scanning and the number of fields/sec for interlaced scanning. All of these formats are permitted in the U.S. digital television standard except the last two formats in bold, which exceed the sample rate constraint of 62.6 Msamples per second.

a complete transition to digital television by 2006 is feasible, eventually the digital television standard will be the standard for terrestrial broadcasting. When digital television becomes more prevalent, the television processing community will surely focus its attention on the *migration path*. The concept of a migration path concerns a future transition from standard HDTV formats to higher resolution formats beyond those in the digital standard. [7, 9] This is the next logical step for terrestrial television broadcasting and was recognized even during the initial stages of the digital standardization process. The desire for the ability to transmit and receive higher resolution video is apparent and the migration path will be an active research area as digital televisions penetrate the market and more bandwidth becomes available. The main goal of this discussion is to begin to understand the issues of the migration path in an effort to develop efficient methods to migrate to higher resolution formats. The conclusions developed can be used to aid a possible future standardization process.

Despite the substantial improvements compared to the NTSC standard, the digital television standard has a significant limitation in its video resolution. A spatial resolution of 1080 x 1920 is the highest spatial resolution with a 16:9 aspect ratio that is permitted in the standard and a temporal resolution of progressive scanning at 60 frames/sec is the highest possible temporal resolution permitted. Note that either the highest spatial or temporal resolution can be achieved separately but the 1080P format (1080 lines with progressive scanning at 60 frames per second) which is the combination of the highest spatial resolution (with a 16:9 aspect ratio) and the highest temporal resolution is not permitted by the standard. The need to be able to transmit and display video in the 1080P format is desired by terrestrial broadcasters since it provides both high spatial and temporal resolution. However, the 1080P format was prohibited from the digital television standard because the standard was intended for transmission over a single 6 MHz terrestrial channel which can currently support approximately 18 Mbps [36] and the pixel rate of the 1080P format is too high for satisfactory compression at this bandwidth.

Although there is much interest in the future transmission of higher resolution formats, the terrestrial transmission of these formats will not be possible unless additional bandwidth becomes available. There are several ways that additional bandwidth may become available such as the allocation of a wider spectrum for each channel and/or improvements may be made in compression or modulation technology. In addition, additional bandwidth may be available immediately or in the near future over media such

as cable or satellite. This thesis will not attempt to speculate when, why or how additional bandwidth will be available, but assumes that there is additional capacity available to implement a migration path and focus on utilizing the bandwidth efficiently to be able to transmit higher resolution formats such as the 1080P format.

5.2.2 Application of Scalable Coding to the Migration Path

One approach to satisfy the demand to broadcast higher resolution formats in the future would be to create another standard separate from the HDTV standard. However, the transition from the NTSC standard to the HDTV standard demonstrated the problems with creating a separate standard and suggest that this should be avoided if possible. The most significant problem was that equipment compliant with the NTSC standard could not be used with the adoption of the new HDTV standard. Note that simulcasting was chosen for the digital television standard mainly because the old NTSC signal that was developed using technology from the 1940's and 1950's was very inefficient. Inclusion of the NTSC signal in the new standard would have made the digital television signal very spectrum inefficient. The signal in the HDTV standard is efficiently compressed and therefore can be used in the migration path. Therefore, it is preferable for the migration path to be *backward-compatible* with the HDTV standard so as not to render earlier digital televisions and equipment obsolete. [37] A backward-compatible migration path would allow higher quality transmissions to be displayed by receivers equipped to handle them, but still permit lower caliber receivers to decode and display standard HDTV formats.

A backward-compatible migration path suggests the use of scalable coding. Scalable coding can also be applied recursively to create multiple levels of service and the capability for multicast broadcasting would be desirable since it is reasonable to expect that resolutions beyond the initial migration will be in demand sometime in the future. Figure 5.1 illustrates an example of a backward-compatible migration path that provides three levels of digital television service, each with increasing spatial resolution.

While a scalable approach can certainly be used to create multiple levels of service, this thesis will only examine a migration scheme with two levels of service. This is done to simplify the analysis and because many of the concepts in a two level scheme can be applied to a multiple level scheme. The base layer should be compliant with the current

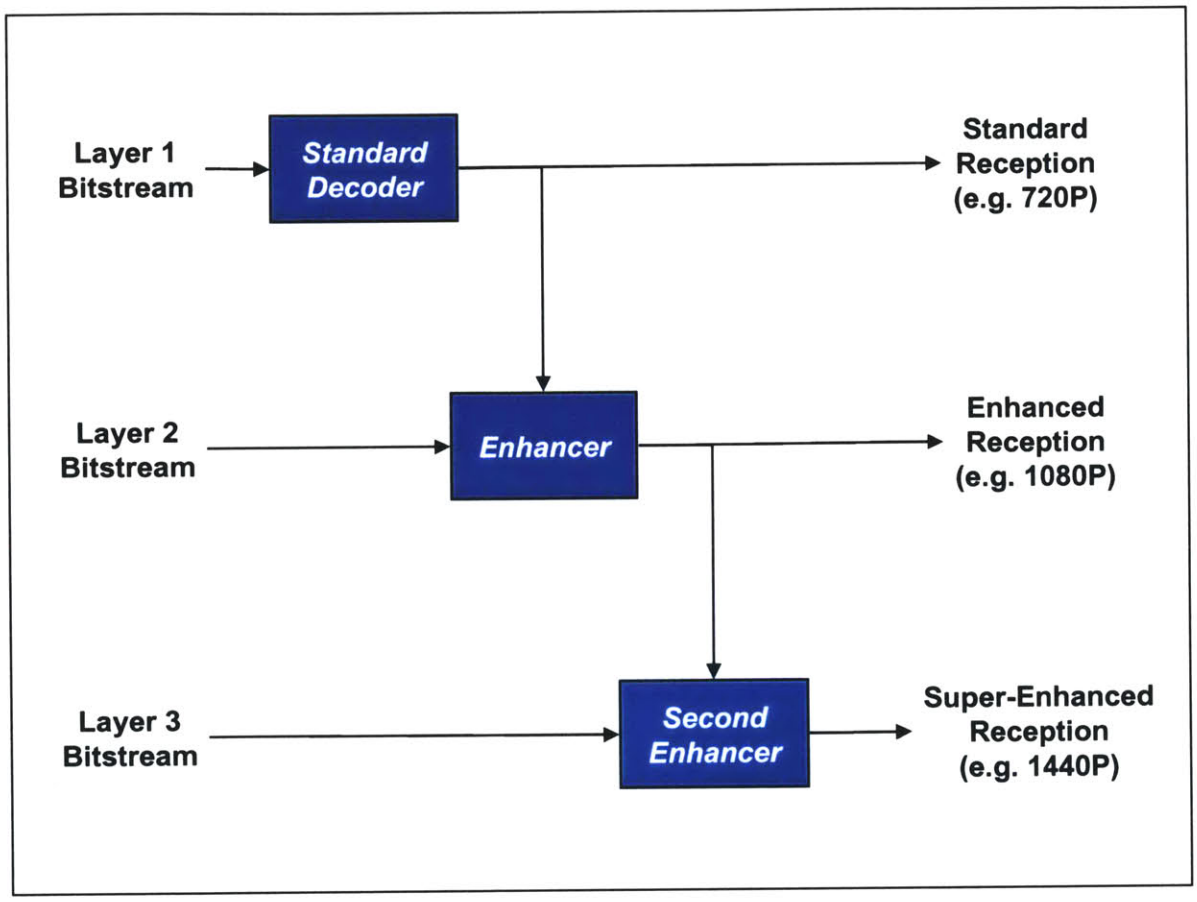

Figure 5.1: Scalable Coding Can Be Used to Achieve Multiple Service Levels. An example of a backward-compatible migration path that provides three levels of digital television service, each with increasing spatial resolution (720P, 1080P and 1440P). The Layer 1 bitstream can be decoded by a standard decoder to provide standard video (720P). An enhancer can utilize the Layer 2 bitstream (in addition to the video generated from the Layer 1 bitstream) to provide reception of enhanced video (1080P). A second enhancer can utilize the Layer 3 bitstream (in addition to the video generated from the Layer 1 and 2 bitstreams) to provide reception of super-enhanced video (1440P).

digital standard but there are no restrictions on the methods to achieve the enhancement since there currently is no syntax governing enhancement layer. This syntax will need to be standardized in the future and this thesis will attempt to provide some results to assist this standardization. There are many different formats that may be chosen for the initial migration, but the 1080P format is an reasonable choice for investigation since there is a demand for combining the highest spatial and temporal resolutions of the current standard. This research will assume that the 1080P format will be the desired enhancement layer for the migration path and focus on developing efficient methods to migrate to this format.

5.2.3 Resolution of the Base Layer and the Migration Path

The enhancement layers in any scalable coding scheme are highly dependent on the base layer. The only requirement of the migration path is for the base layer to be compliant with the HDTV standard to ensure backward-compatibility. However, unlike many other scalable coding scenarios where the format of the base layer is fixed, the migration path is more complicated since the standard permits multiple formats. It will be shown that different signal processing operations will have to be performed to construct the desired 1080P enhancement layer for different base layers. Thus, the migration path for different base layers should be examined independently.

Terrestrial broadcasters have not reached a consensus on a single broadcasting format that will be used for HDTV, but most have decided on either the 1080I or the 720P format. (Note that if future research were to show that one format was much better for future migration, it might convince broadcasters to agree on a single format.) The 1080I and 720P formats are ideally suited for the migration path since their resolutions are related to the 1080P format by simple integer ratios. Video in the 1080I format has $\frac{1}{2}$ the number of lines and the same number of pixels per line as the 1080P format. Deinterlacing of the 1080I video can be performed to create the desired 1080P enhancement layer. Video in the 720P format has $\frac{2}{3}$ of both the lines of resolution and pixels per line as the 1080P format. Spatial upsampling of the 720P video can be performed to create the desired 1080P enhancement layer. Since deinterlacing and spatial upsampling are very different signal processing operations, it is clear that each case should be handled differently.

There is another logical migration scheme that would involve a different base layer besides 720P and 1080I. The 1080P@30fps format is another base layer format that is related to the 1080P format by a simple ratio of integers. Video in the 1080P@30fps format has $\frac{1}{2}$ the number of frames as the 1080P format. Temporal upsampling of the 1080P@30fps video can be performed to create the desired 1080P enhancement layer. However, this version of the migration path will not be discussed or studied in this thesis because it appears that the 1080P@30fps format will not have much practical significance since most digital television broadcasters are planning on using either the 720P or 1080I format. The avoidance of the 1080P@30fps format by broadcasters is mainly caused by the fact that most displays will not be in that format. Thus, additional format conversion will need to performed prior to display and this additional processing may result in additional artifacts. Figure 5.2 illustrates the different signal processing operations that the migration path would have to implement to migrate from different base layers to the desired 1080P enhancement layer format.

The migration path schemes involving the 1080I and 720P formats as the base layer will be discussed in detail in the next section. Note that the focus will be on the 1080I case since the experimental results performed in the previous chapter examined that case, but one can easily see how similar conclusions can be assumed for the 720P case (as well as the 720P@30fps case).

5.3 Role of Adaptive Format Conversion in the Migration Path

There are many scenarios where the use of adaptive format conversion as enhancement data may be beneficial. The migration path for digital television [9] is one application where adaptive format conversion may become very important. This section discusses both the initial and future roles of adaptive format conversion in the migration path. Figure 5.3 illustrates an overview of the application of adaptive format conversion to the migration path for digital television. In this figure, a standard decoder decodes video in a format that is part of the current standard such as 720P or 1080I. This format will be the base layer for scalable coding and will significantly affect the enhancement

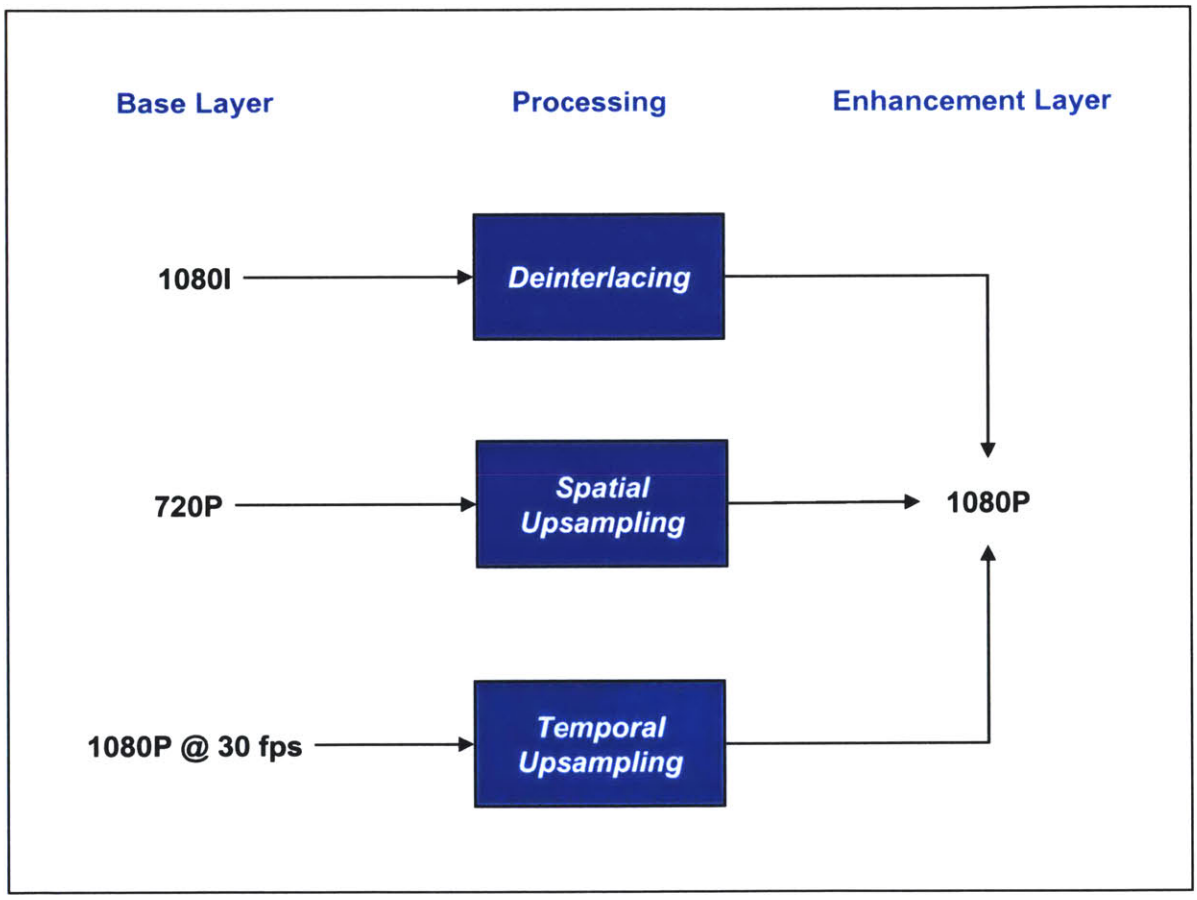

Figure 5.2: Resolution of the Base Layer Format and the Migration Path. Different signal processing operations have to be performed for different base layers to achieve the 1080P enhancement layer format: deinterlacing must be performed for the 1080I format, spatial upsampling must be performed for the 720P format and temporal upsampling must be performed for the 1080P@30fps format.

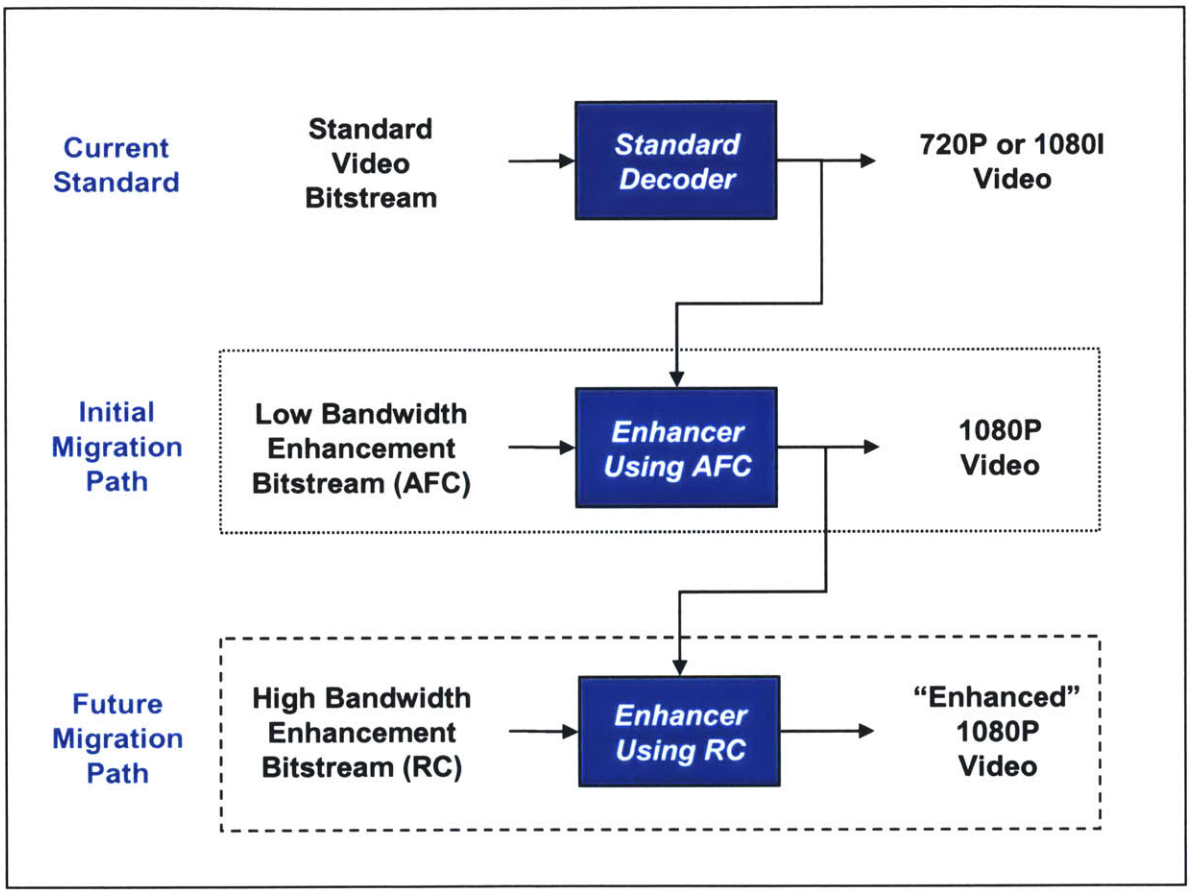

Figure 5.3: Role of Adaptive Format Conversion in the Migration Path. The initial migration path would involve a low bandwidth enhancement bitstream containing adaptive format conversion (AFC) information. This allows creation of the desired 1080P format. Future migrations are not limited by the initial migration. A future migration using a high bandwidth enhancement bitstream containing residual coding (RC) information is shown in this example. The additional enhancement data can be used to create an "enhanced" version of the 1080P video.

data transmitted as discussed in the previous section. The initial migration path would involve an enhancer that utilizes a low bandwidth enhancement bitstream which contains adaptive format conversion information. The adaptive format conversion information would be used in addition to the decoded base layer video from the standard decoder to construct video in the 1080P format. Note that an initial migration which uses adaptive format conversion information does not limit future migrations. If additional bandwidth were to become available in the future, another enhancer can utilize a high bandwidth enhancement bitstream containing residual coding information to further enhance the 1080P video. This figure illustrates how easily adaptive format conversion could be applied to the migration path problem.

5.3.1 Initial Role of Adaptive Format Conversion in the Migration Path

Scalable coding provides backward-compatible scalability, but an additional constraint of the migration path is that the bandwidth for any enhancement layer is expected to be small in the near future. Transmission bandwidth is very expensive due to the immense demand for available spectrum by an increasing number of applications. In fact, one reason that the FCC wants to reclaim the spectrum currently used for analog terrestrial broadcasts (after digital television becomes more widespread) is to resell the spectrum. A limited amount of enhancement bandwidth is very significant because it discourages residual coding, the typical form of enhancement data used in most scalable coding schemes. As discussed earlier, residual coding can recover (practically) all of the video detail in the enhancement layer, albeit this may require use of a very fine quantizer to code the prediction error which will result in a high enhancement bitrate. At low enhancement bitrates, it may not be possible to achieve the target bitrate using residual coding, even with the coarsest quantizer. Furthermore, even if the target bitrate is achieved with a coarse quantizer, a coarsely coded residual often will not significantly improve the video quality of the enhancement layer.

The experimental results in Section 4.4 that compared adaptive format conversion and residual coding can be applied to a migration path with a low enhancement bandwidth. Adaptive format conversion was shown to provide scalability at low enhancement bitrates that is not possible with residual coding. It is currently unclear how much additional bandwidth will be available in the future, but one estimate is that an additional

two to three Mbps will be available for the migration path of digital television. One method of applying the experimental results in this thesis (which were performed with sequences of CIF resolution) to the migration path problem (which involves sequences of HDTV resolution) is to normalize bitrates by using the bits per pixel metric (BPP). Since 1 Mbps $= 1048576$ bits/sec and

$$\frac{1048576 \text{ bits/sec}}{(1080 * 1920 \text{ pixels/frames})(60 \text{ frames/sec})} = 0.0084 \text{ bits/pixel}, \qquad (5.1)$$

the estimated two to three Mbps of available enhancement bitrate would represent an additional 0.0168 to 0.0252 BPP that could be used to enhance the enhancement layer. Looking at this region in Figure 4.7 shows that only adaptive format conversion can be used for enhancement data since residual coding will result in an enhancement bitrate greater than the available bandwidth even with the coarsest quantizer. Achieving video scalability using only residual coding for enhancement data cannot occur unless at least approximately 0.05 BPP of enhancement bitrate (equivalent to 6 Mbps) are available. Therefore, one can implement the migration path with a smaller amount of available bandwidth using adaptive format conversion information. The video scalability provided by adaptive format conversion is useful, but even more important is the coding gains that are achieved even at these low enhancement bitrates. This was demonstrated in Section 4.4 which not only showed that there can be coding gains, but the coding gains can be quite substantial when the base layer is coded well. Digital television signals will definitely be coded with a high fidelity, therefore, adaptive format conversion will be able to use accurate information from the base layer and produce significant improvement in the video quality of the enhancement layer.

These results indicate that adaptive format conversion is an ideal choice to provide video scalability for the migration path in the near future. The ability of adaptive format conversion to provide efficient video scalability at low enhancement bitrates not only matches the initial constraint on the enhancement layer bandwidth, but also takes advantage of the high fidelity of the base layer to provide significant coding gains. The signal processing required to implement an initial migration path using only adaptive format conversion information is shown in Figures 5.4 and 5.5 when the base layer is in the 1080I and 720P formats, respectively. These figures explicitly define all the elements needed in both the transmitter and the receiver. The elements in the dotted boxes

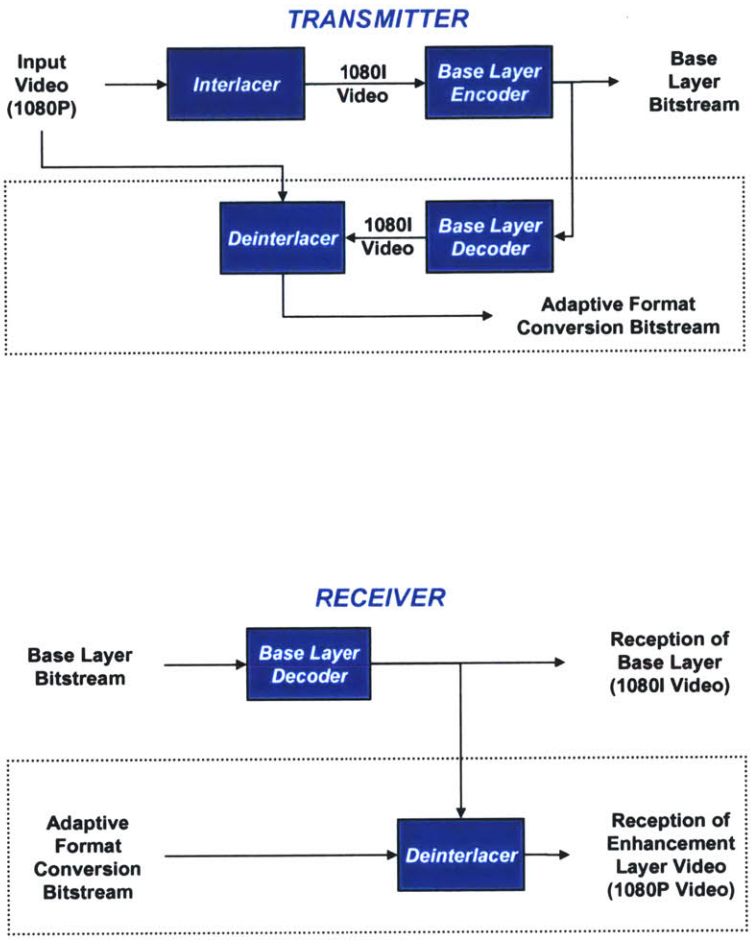

Figure 5.4: An Initial Migration Path System with 1080I as the Base Layer Format. If the elements in the dotted boxes are ignored, what remains is a standard HDTV system. The 1080P input video is interlaced to create 1080I video (an allowable transmission format) that is encoded and transmitted as the base layer. At the receiver, the base layer bitstream can be decoded to receive 1080I video, thus, backward-compatibility is achieved. The elements in the dotted boxes implement the migration path. The transmitter mimics a standard video decoder and decodes the base layer bitstream. The decoded 1080I video is used with the 1080P input video to generate an enhancement layer. The enhancement data consists of adaptive format conversion information and can be used by a deinterlacer in an advanced receiver to receive video in the 1080P format.

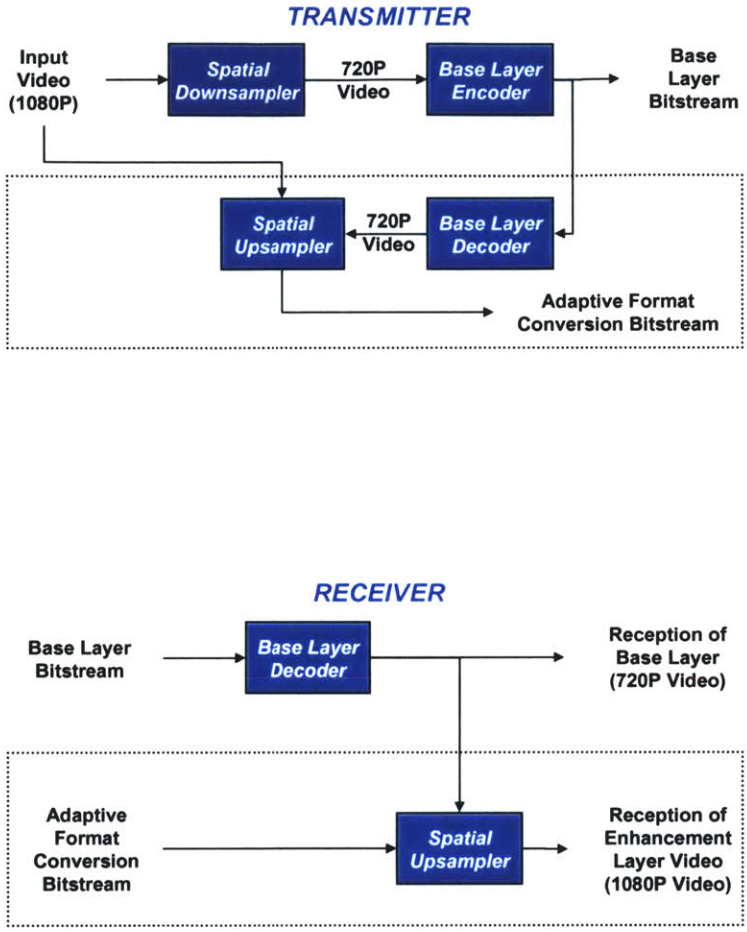

Figure 5.5: An Initial Migration Path System with 720P as the Base Layer Format. If the elements in the dotted boxes are ignored, what remains is a standard HDTV system. The 1080P input video is spatially downsampled to create 720P video (an allowable transmission format) that is encoded and transmitted as the base layer. At the receiver, the base layer bitstream can be decoded to receive 720P video, thus, backward-compatibility is achieved. The elements in the dotted boxes implement the migration path. The transmitter mimics a standard video decoder and decodes the base layer bitstream. The decoded 720P video is used with the 1080P input video to generate an enhancement layer. The enhancement data consists of adaptive format conversion information and can be used by a spatial upsampler in an advanced receiver to receive video in the 1080P format.

represent the components that would need to be added to the current HDTV system to support the use of adaptive format conversion information. If the elements in the dotted boxes are ignored, what remains is a standard HDTV system.

Figure 5.4 shows the migration path with the 1080I format as the base layer. The input video is in the 1080P format and must first be converted to the 1080I format at the transmitter before transmission. The 1080I video is compressed and transmitted as the base layer. In addition to encoding the 1080I video, the transmitter mimics a standard receiver and decodes the 1080I video. The decoded 1080I video can be converted to the 1080P format by deinterlacing and enhancement data in the form of adaptive deinterlacing information is transmitted in parallel with the base layer bitstream. At the receiver side, a standard receiver can ignore the enhancement bits and decode only the base layer bitstream to produce 1080I video. Thus, standard reception is unaffected and backward-compatibility is achieved. An advanced receiver utilizes the adaptive format conversion information from the enhancement layer bitstream along with the decoded 1080I base layer to receive 1080P video.

Alternatively, the 720P format can be the base layer for the migration path as shown in Figure 5.5. In this case, the 1080P input video is converted to the 720P format prior to compression and transmission as the base layer. Note that the ratio between the spatial resolution of the formats is $\frac{2}{3}$, therefore, format conversion can be achieved by spatially upsampling by two and then downsampling by three. This combined operation is referred to as spatial downsampling in the figure since the total data rate is reduced. In addition to encoding the 720P video, the transmitter mimics a standard receiver and decodes the 720P video. The decoded 720P video can be converted to the 1080P format by spatial upsampling and used to generate enhancement data that is transmitted in parallel with the base layer bitstream. Note that this format conversion can be achieved by spatial upsampling by three and then downsampling by two. The combined operation is referred to as spatial upsampling in the figure since the total data rate is increased. At the receiver side, a standard receiver can ignore the enhancement bits and decode only the base layer bitstream to produce 720P video. Thus, standard reception is unaffected and backward-compatibility is achieved. An advanced receiver utilizes the adaptive format conversion information from the enhancement layer bitstream along with the decoded 720P base layer to receive 1080P video.

5.3.2 Future Role of Adaptive Format Conversion in the Migration Path

In the previous section, it was shown that adaptive format conversion was an ideal solution for a migration path when the enhancement bandwidth is small. In the future, more bandwidth may be available to the migration path to support additional levels of scalability. This bandwidth could be used to further enhance the quality of the 1080P picture achieved with the initial migration path. Additional bandwidth could also be used to migrate to higher resolution formats beyond 1080P. These two alternatives are shown in Figure 5.6. This section will focus on the first choice and a straightforward method to accomplish this is to transmit residual coding information to further enhance the 1080P video that was created using adaptive format conversion information. Note that the second choice of migrating to formats beyond 1080P can be achieved by recursively applying concepts similar to those used for the initial migration path. For example, adaptive spatial upsampling could be used to migrate from 1080P to 1440P and adaptive temporal upsampling could be used to migrate from 1080P to 1080P@72fps (1080 lines with progressive scanning at 72 frames/second).

Note that our current scalable coding scheme uses residual coding information "on top of" adaptive format conversion information to create "enhanced" 1080P video. It is natural to examine whether adaptive format conversion information is still efficient when there is enough bandwidth to support residual coding. If the use of adaptive format conversion information is not efficient at higher enhancement bitrates, it would be more efficient to migrate from a standard HDTV format to 1080P by using nonadaptive format conversion and residual coding. This analysis is accomplished by comparing the use of both types of enhancement data (adaptive format conversion and residual coding) to the use of residual coding (after nonadaptive format conversion). The experiments in Section 4.5 showed that the use of both types of enhancement data is more efficient than using only residual coding data when the base layer is coded well as it is in digital television. Therefore, the implementation of an initial migration path using adaptive format conversion information not only provides scalability in the short-term, but also provides coding gains in the future after a second migration path is implemented using residual coding information.

The signal processing required to implement a migration path using both adaptive

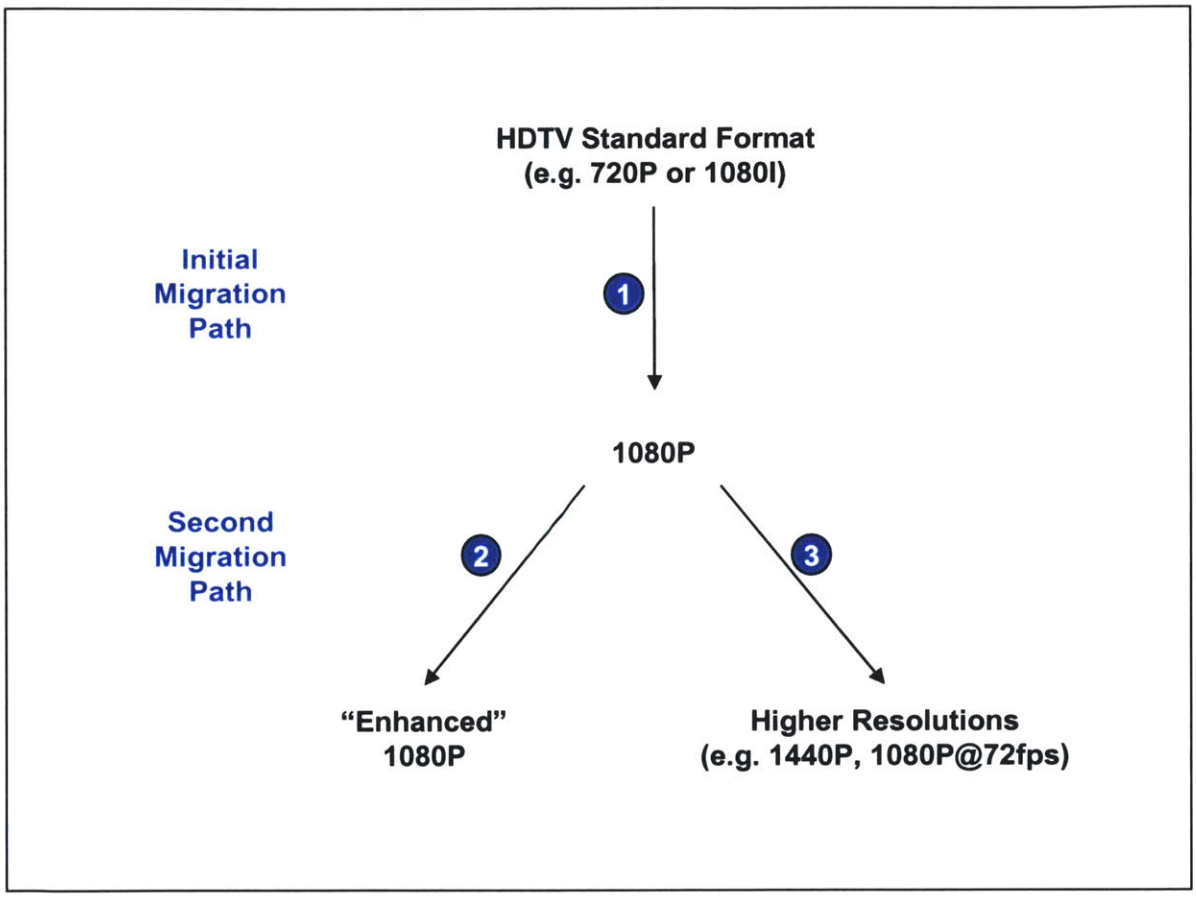

Figure 5.6: Possibilities After An Initial Migration Path Using Adaptive Format Conversion Information. In this example, an initial migration path (1) was implemented using adaptive format conversion information to migrate from a HDTV standard format (such as 720P or 1080I) to the 1080P format which is not in the current standard. In the future, more bandwidth may become available to the migration path. A second migration path could be implemented to either further enhance reception of 1080P video (2) or to reach higher resolution formats such as 1440P or 1080P@72fps (3).

format conversion and residual coding information is shown in Figures 5.7 and 5.8 when the base layer format is in the 1080I and 720P formats, respectively. The first migration path will use only adaptive format conversion information for enhancement data as described in the previous section. The second migration path will use residual coding for enhancement data. These figures explicitly define all the elements needed in both the transmitter and receiver. The elements in the dotted boxes represent the components that were added to the current HDTV system to support the initial migration and the elements in the dashed boxes need to be added to the current HDTV system to support residual coding. If the elements in both the dotted and dashed boxes are ignored, what remains is a standard HDTV system.

Figure 5.7 shows the migration path with the 1080I format as the base layer. Note that this figure is similar to Figure 5.4 except for the components listed in the boxes with dashed lines. In addition to composing the base layer and adaptive format conversion information for the first enhancement layer in the same manner as in Figure 5.4, the transmitter also has to compute the residual between the original 1080P input video and the 1080P video created after deinterlacing the 1080I video. The coded residual is then transmitted as the second enhancement layer. A receiver can utilize the additional information from the decoded residual to further enhance the 1080P video. Note that backward-compatibility is achieved and three different types of reception are possible: standard reception (1080I), advanced reception (1080P) and super-advanced reception (enhanced 1080P).

Figure 5.8 shows the migration path with the 720P format as the base layer. Note that this figure is similar to Figure 5.5 except for the components listed in the boxes with dashed lines. In addition to composing the base layer and adaptive format conversion information for the first enhancement layer in the same manner as in Figure 5.5, the transmitter also has to compute the residual between the original 1080P input video and the 1080P video created after spatial upsampling the 720P video. The coded residual is then transmitted as the second enhancement layer. A receiver can utilize the additional information from the decoded residual to further enhance the 1080P video. Note that backward-compatibility is achieved and three different types of reception are possible: standard reception (720P), advanced reception (1080P) and super-advanced reception (enhanced 1080P).

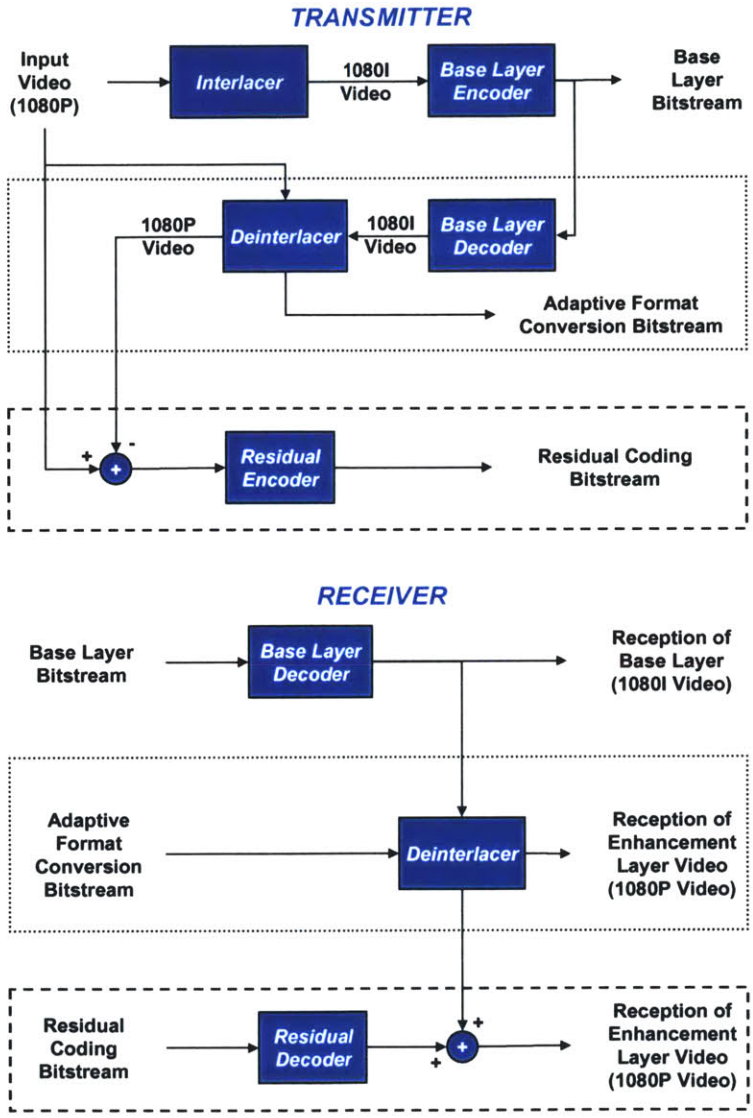

Figure 5.7: A Future Migration Path System with 1080I as the Base Layer Format. The dotted boxes represent an initial migration path which uses adaptive format conversion information as enhancement data. The dashed boxes represent a second migration path which uses residual coding information as enhancement data. If the elements in the dotted and dashed boxes are ignored, what remains is a standard HDTV system. The only elements that differ from those in Figure 5.4 are the elements in the dashed boxes which implement the second migration path. The transmitter can compute the residual between the original 1080P video and the 1080P video created after deinterlacing the 1080I video. This residual can be coded and transmitted as another enhancement layer. A receiver may decode this residual to further enhance the reception of 1080P video.

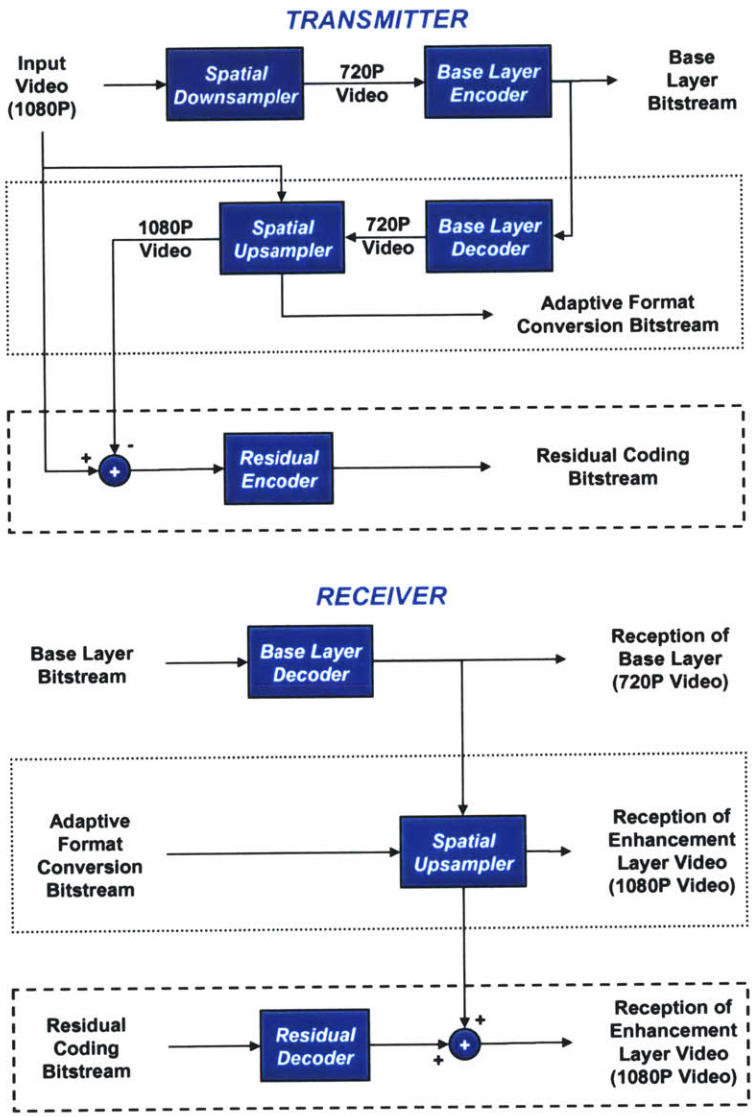

Figure 5.8: A Future Migration Path System with 720P as the Base Layer Format. The dotted boxes represent an initial migration path which uses adaptive format conversion information as enhancement data. The dashed boxes represent a second migration path which uses residual coding information as enhancement data. If the elements in the dotted and dashed boxes are ignored, what remains is a standard HDTV system. The only elements that differ from those in Figure 5.5 are the elements in the dashed boxes which implement the second migration path. The transmitter can compute the residual between the original 1080P video and the 1080P video created after spatial upsampling the 720P video. This residual can be coded and transmitted as another enhancement layer. A receiver may decode this residual to further enhance the reception of 1080P video.

5.4 Similar Applications

This thesis focused on applying its results to terrestrial broadcasting. However, this work can have significant impact on other environments such as cable and satellite transmissions. Terrestrial channel capacities may limit the immediate applicability of a migration path, however environments such as cable and satellite may be able to implement a migration scheme in the near future. In these environments, a 6 MHz channel may be able to support higher capacities than the 18 Mbps bandwidth of a terrestrial channel. These environments also may not be limited to using a single 6 MHz channel for each transmission.

In addition to migration with HDTV formats, another area where this thesis may be applicable is migration for *standard-definition television* (SDTV). Many consumers will wait for the price of digital televisions to decrease and purchase set-top boxes to display digital content on analog televisions. This may cause many digital broadcasts to be in the 480I format (480 lines of resolution with interlaced scanning at 60 fields/sec). Using the same general concept as the migration path from 1080I to 1080P video, a migration path can be utilized to assist the display of interlaced SDTV material on progressive displays such as computer monitors, i.e. the migration from 480I to 480P (480 lines of resolution with progressive scanning at 60 frames/sec).

5.5 Summary

This chapter began by reviewing the U.S. digital television standard. The standard has many substantial improvements over its analog counterpart. However, the standard has limitations on the transmittable video formats and the need to migrate to higher resolutions in the future has already been recognized. The concept of a migration path concerns the transition to resolutions beyond the current standard in a backward-compatible manner so as not to render earlier digital televisions obsolete. Scalable coding provides backward-compatible scalability, but an additional constraint of the migration path is that the bandwidth for any enhancement layer is expected to be low in the near future which discourages residual coding. The results in this thesis indicate that adaptive format conversion is an ideal choice to provide video scalability for the migration path in the short-

term as well as the long-term. The ability of adaptive format conversion to provide efficient video scalability at low enhancement bitrates matches the short-term requirement of the migration path that the enhancement layer bandwidth is low. Adaptive format conversion information is also coding efficient at higher enhancement bitrates when the base layer is coded well. Since the base layer of digital television will be coded at a high fidelity, adaptive format conversion information will also be beneficial to the migration path when more bandwidth is available in the future to support residual coding.

Chapter 6

Conclusions

6.1 Summary

A common problem for many video applications is that multiple clients require different types of service due to variations in their available bandwidth, processing power and/or memory resources. Scalable coding is a popular technique to provide multiple levels of video service since enhancement layers can efficiently exploit redundant information in previous layers. Despite the fact that scalable coding is well known in the video compression field and been implemented in many coding standards, the use of adaptive format conversion information as enhancement data has often been overlooked and is not well understood. This thesis begins by reviewing the previous research of Sunshine which demonstrated that a significant improvement in video quality could be obtained with a small amount of enhancement data to assist adaptive deinterlacing at the decoder. The simulation results were obtained without coding the base layer and suboptimal parameter selection, but demonstrate the potential of using adaptive format conversion information as enhancement data. It is clear that adaptive format conversion must be researched further and this thesis investigates when and how adaptive format conversion could be used to improve scalable video compression.

First, a new scalable codec was developed that can utilize adaptive format conversion information and/or residual coding information as enhancement data. The ability to utilize either or both types of enhancement data in simulations is necessary to better understand adaptive format conversion. First, it permits comparison between adaptive format conversion and residual coding when each is transmitted independently. Second, one does not need to choose between the two types of enhancement data and both can be transmitted when there is enough enhancement layer bandwidth. The use of both adaptive format conversion and residual coding has not previously been studied and this

codec permits such experiments. The implementation described in this thesis also includes a base layer codec to investigate the effect of the base layer and an algorithm for optimal mode selection when variable-sized block partitioning is performed. This codec was then used to perform various simulations to investigate different aspects of adaptive format conversion.

The first experiments examined adaptive format conversion on the base layer after it was coded at a wide range of qualities to examine the effect of base layer coding on adaptive format conversion. Since previous simulations used an uncoded base layer which provides perfect information from the remaining fields to reconstruct the missing field, it was important to investigate this issue. Adaptive format conversion proved to be useful over a wide range of base layer qualities. As expected, large gains were seen when the base layer was coded well. In addition, PSNR gains over 1 dB were present even when the base layer was coded poorly. Thus, adaptive format conversion was concluded to be an efficient type of enhancement information and can significantly improve the quality of the decoded enhancement layer even when the base layer is not coded at a high fidelity.

Another experiment compared adaptive format conversion and residual coding. Since adaptive format conversion usually involves a smaller number of coded parameters per region compared to residual coding, it can provide video scalability at low enhancement layer bitrates (between 0.01 BPP and 0.05 BPP) that are not attainable with residual coding. The experimental results also showed that adaptive format conversion is usually preferable at medium enhancement layer bitrates (between 0.05 BPP and 0.12 BPP) when the use of either type of enhancement data is possible. Residual coding only outperforms adaptive format conversion at these enhancement layer bitrates when the base layer is coded poorly.

One research area that has not been previously investigated is the use of both adaptive format conversion information and residual coding information as enhancement data. It was previously unclear whether the use of adaptive format conversion would help or hinder video scalability at high enhancement layer bitrates. This was investigated by comparing the use of both types of enhancement data to the use of only residual coding. Experimental results showed that the use of adaptive format conversion usually improves coding efficiency at higher enhancement layer bitrates. The use of adaptive format conversion information reduces the coding efficiency only when the base layer is coded poorly.

The last chapter investigated the application of adaptive format conversion to the migration path for terrestrial digital television in the United States. The results presented in this thesis indicate that adaptive format conversion is well-suited to handle the unique problems of the migration path. The initial difficulty with a migration path structure is the limited bandwidth that would be available to support the enhancement layer. The limited bandwidth discourages residual coding, but is well matched to adaptive format conversion which can provide efficient scalability at low enhancement bitrates. Therefore, an initial migration path could be implemented using only adaptive format conversion information as enhancement data. Adaptive format conversion information can also assist a future migration path when more bandwidth is available to support the additional transmission of residual coding information as enhancement data. Since television signals are coded with high fidelity, the use of both adaptive format conversion and residual coding will be more efficient than using only residual coding. Therefore, the implementation of an initial migration path does not hinder and actually improves a future migration path.

Most of this thesis was focused on a special case of adaptive format conversion: adaptive deinterlacing for the migration path problem. It should be noted that many of the same concepts and conclusions can also be applied to other cases and applications. For example, adaptive spatial upsampling will require different signal processing modes than adaptive deinterlacing but the structure and parameter selection of a codec to perform adaptive spatial upsampling will probably be very similar to the one described in this thesis. Other applications besides the migration path for terrestrial digital television may also benefit from this work since adaptive format conversion can be applied to practically any scalable video coding algorithm.

6.2 Future Research Directions

The results of this thesis suggest that adaptive format conversion could play a major role in the migration path to higher resolutions for terrestrial digital television systems. Interest in the migration path will continue to increase as digital television becomes more prevalent. Additional development on the application of adaptive format conversion to this problem must be done to get closer to a practical implementation. This research

addressed some of the limitations of previous work in the field such as the effect of base layer coding and optimal parameter selection. In addition, this thesis examined the use of both adaptive format conversion information and residual coding information for video scalability at high enhancement bitrates. However, other issues need to be investigated to further substantiate the use of adaptive format conversion for the migration path. This research was performed on CIF resolution sequences and the bitrates were normalized using the bits per pixel metric to apply the experimental results to the migration path problem. A possible avenue of future research is to perform experiments with high definition video formats and the corresponding bitrates. The experiments could also be performed on a wider range of video sequences to provide more support to a proposal for a possible migration path structure.

Another area for future research is to investigate the effect of the resolution format of the base layer. In this thesis, the resolution format of the base layer was chosen to be interlaced video with the same spatial resolution as the progressive enhancement layer. Different base layers can be used to "migrate" to a specific enhancement layer and this thesis has also discussed the use of adaptive spatial upsampling to migrate from the 720P format to the 1080P format. Adaptive spatial upsampling from the 720P format is conceptually very similar to the adaptive deinterlacing case examined in this thesis, but the different format conversions will require different video processing techniques. It is currently unclear whether the use of one base layer format will work significantly better than another format in a scalable coding scheme with the same enhancement layer.

Two issues need to be carefully considered to permit a fair comparison. The first issue is the quality of the base layer. Even though the base layers of two different migration schemes may be coded at the same bitrate, comparison of the base layer video sequences is difficult because they will be in different formats. In addition, it is known that coding interlaced video is less efficient than coding progressive video. [38] These issues make it difficult to compare the base layers of different migration schemes. The enhancement layer in the different schemes will be the same so they can be directly compared using a measure such as the PSNR. Some insight to this problem can be gained by briefly considering the migration path when the base layer is not coded. The 1080I scheme will perfectly reconstruct the 1080P video in stationary regions since the missing line can be repeated from the previous or subsequent field. On the other hand, the 720P scheme will not be able to replace the detail lost by the spatial downsampling performed at the encoder to

generate the base layer. This illustrates a situation where the spatio-temporal tradeoff of interlaced coding is not apparent. However, when there is motion in the video, it would be interesting to determine whether interlace artifact or subsampling is more detrimental to reconstruction of the original video and this issue will be further complicated when the base layer is compressed. The second issue to consider in a comparison between the 720P and 1080I migration paths is whether both schemes are using efficient methods to enhance the base layer. For example, if the deinterlacing algorithms are more efficient than the spatial upsampling techniques, it may not be fair to compare the results of the enhancement layer. Thus, the efficiency of enhancement techniques for both scenarios must be examined. These two issues must be resolved for a fair comparison between the 720P and 1080I migration paths.

Another future research direction is to investigate the allocation of channel bandwidth between the different layers of a migration path. In this thesis, the bandwidth issue was addressed by assuming the base layer was given, i.e. the base layer bitrate and distortion were already determined, and the problem was "reduced" to examining the enhancement layer quality as a function of the enhancement layer bitrate. A different problem formulation is a constraint on the total bitrate of the base and enhancement layers. This scenario is more representative of practical applications and must be better understood before a migration path can be realistically implemented. This constraint requires one to examine the quality of both layers as a function of the base layer bitrate (the enhancement layer bitrate is fixed once the base layer bitrate is determined since the total bitrate is fixed). For example, assume that 20 Mbps are available for both layers of a migration path scheme. In this thesis, it was assumed that 18 Mbps would be used for the base layer and the goal was to maximize the enhancement layer quality using the remaining 2 Mbps for enhancement data. It is unclear whether it would be better to choose a different bandwidth allocation such as 17 Mbps for the base layer and 3 Mbps for the enhancement layer or any of many other bandwidth allocations. The current practice is to allocate bandwidth based on the relative pixel rate of the layers without considering the content of the material. Bandwidth allocation should probably depend on the video sequence to be coded. This problem is of great interest, but it is currently unclear how to investigate this problem. Note that the bandwidth allocation problem is not specific to the use of adaptive format conversion as enhancement data. The general problem of allocating total bandwidth to different layers of a scalable coding scheme is not well

understood, so any insight gained while investigating adaptive format conversion would probably also be applicable to many other scalable coding scenarios.

Bibliography

[1] A. Lallet, C. Dolbear, J. Hughes, and P. Hobson, "Review of Scalable Video Strategies for Distributed Video Applications," in *Distibuted Imaging, IEE European Workshop*, vol. 1999, (London, UK), pp. 2/1–2/7, IEE, November 1999.

[2] I. JTC1/SC29/WG11, "13818-2: MPEG-2 Video," tech. rep., ISO/IEC, 1995.

[3] B. Haskell, A. Puri, and A. Netravali, eds., *Digital Video: An Introduction to MPEG-2*. Digital Multimedia Standard Series, New York, NY, USA: Chapman and Hill, 1997.

[4] I. JTC1/SC29/WG11, "14496-2: MPEG-4 Video," tech. rep., ISO/IEC, 1999.

[5] W. Wan and J. Lim, "Adaptive Format COnversion For Video Scalability at Low Enhancement Bitrates," in *MWSCAS 2001: Proceedings of the 44th IEEE 2001 Midwest Symposium on Circuits and Systems*, vol. 2, (Fairborn, OH), pp. 588–592, IEEE, August 2001.

[6] W. Wan and J. Lim, "Adaptive Format Conversion For Scalable Video Coding," in *Proceedings of SPIE: Applications of Digital Image Processing XXIV*, vol. 4472, (San Diego, CA, USA), pp. 390–401, SPIE, July 2001.

[7] J. Lim and L. Sunshine, "HDTV Transmission Formats and Migration Path," *International Journal of Imaging Systems and Technology*, vol. 5, pp. 286–291, Winter 1994.

[8] L. Sunshine, *HDTV Transmission Format Conversion and Migration Path*. Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA, September 1997.

[9] J. Lim, "A Migration Path to a Better Digital Television System," *SMPTE Journal*, vol. 103, pp. 2–6, January 1994.

[10] Y. Faroudja, "NTSC and Beyond," *IEEE Transactions on Consumer Electronics*, vol. 34, pp. 166–178, February 1988.

[11] J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," *IEEE Transactions on Information Theory*, vol. IT-23, pp. 337–343, May 1977.

[12] J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-Rate Coding," *IEEE Transactions on Information Theory*, vol. IT-24, pp. 530–536, September 1978.

[13] I. JTC1/SC29/WG11, "11172-2: MPEG-1 Video," tech. rep., ISO/IEC, 1993.

[14] J. Mitchell, W. Pennebaker, C. Fogg, and D. LeGall, eds., *MPEG Video Compression Standard*. Digital Multimedia Standard Series, New York, NY, USA: Chapman and Hill, 1997.

[15] T. Ebrahimi and C. Horne, "MPEG-4 Natural Video Coding - An Overview," *Signal Processing: Image Communication*, vol. 15, pp. 365–385, January 2000.

[16] K. Rao and P. Yip, *Discrete Cosine Transform. Algorithms, Advantages, Applications*. San Diego, CA, USA: Academic Press, 1990.

[17] T. Chiang and D. Anastassiou, "Hierarchical Coding of Digital Television," *IEEE Communications Magazine*, vol. 32, pp. 38–45, May 1994.

[18] M. Vetterli and K. Uz, "Multiresolution Coding Techniques for Digital Television: A Review," *Multidimensional Systems & Signal Processing*, vol. 3, pp. 161–187, May 1992.

[19] K. Ramchandran, A. Ortega, K. M. Uz, and M. Vetterli, "Multiresolution Broadcast for Digital HDTV Using Joint Source/Channel Coding," *IEEE Journal on Selected Areas in Communication*, vol. 11, pp. 6–23, January 1993.

[20] A. Puri and A. Wong, "Spatial Domain Resolution Scalable Video Coding," in *Proceedings of SPIE - the International Society for Optical Communication*, vol. 2094, (Cambridge, MA, USA), pp. 718–729, SPIE, November 1993.

[21] A.Puri, L. Yan, and B. Haskell, "Temporal Resolution Scalable Video Coding," in *ICIP-94: International Conference on Image Processing 1994*, vol. 2, (Austin, TX, USA), pp. 947–951, IEEE Signal Processing Society, November 1994.

[22] H. Sun and W. Kwok, "MPEG Video Coding with Temporal Scalability," in *ICC '95: 1995 International Conference on Communications*, vol. 3, (Seattle, WA, USA), pp. 1742–1746, IEEE Communication Society, June 1995.

[23] W. Wan, X. Chen, and A. Luthra, "Video Compression for Multicast Environments Using Spatial Scalability and Simulcast Coding," *International Journal of Imaging Systems and Technology*, Submitted.

[24] W. Li, F. Ling, and X. Chen, "Fine Granularity Scalability in MPEG-4 for Streaming Video," in *ISCAS 2000 - International Symposium on Circuits and Systems*, vol. 1, (Geneva, Switzerland), pp. 299–302, IEEE, May 2000.

[25] G. D. Haan and E. Bellers, "Deinterlacing - An Overview," *Proceedings of the IEEE*, vol. 86, pp. 1839–1857, September 1998.

[26] D. Martinez and J. Lim, "Spatial Interpolation of Interlaced Television Pictures," in *ICASSP-89: 1989 International Conference on Acoustics, Speech and Signal Processing*, vol. 3, (Glasgow, UK), pp. 1886–1889, IEEE, May 1989.

[27] A. Ortega and K. Ramchandran, "Rate-Distortion Methods for Image and Video Compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 23–50, November 1998.

[28] G. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74–90, November 1998.

[29] A. T. S. Committee, "ATSC Digital Television Standard," tech. rep., ATSC, September 16, 1995.

[30] F. C. Commision, "Fourth Report and Order," Tech. Rep. 96-493, FCC, December 27, 1996.

[31] C. Basile *et al.*, "The US HDTV Standard," *IEEE Spectrum*, vol. 32, pp. 36–45, April 1995.

[32] E. Petajan, "The HDTV Grand Alliance System," *IEEE Communications Magazine*, vol. 34, pp. 126–132, June 1996.

[33] J. Lim, "Digital Television: Here At Last," *Scientific American (International Edition)*, vol. 278, pp. 56–61, May 1998.

[34] D. Strachan, "HDTV in North America," *SMPTE Journal*, vol. 105, pp. 125–129, March 1996.

[35] D. Strachan, "Video Compression," *SMPTE Journal*, vol. 105, pp. 68–73, February 1996.

[36] Y. Wu and B. Caron, "Digital Television Terrestrial Broadcasting," *IEEE Communications Magazine*, vol. 32, pp. 46–52, May 1994.

[37] W. Schreiber, M. Polley, and S. Wee, "Digital Television Broadcasting: Nondisruptive Improvement Over Time," *SMPTE Journal*, vol. 106, pp. 439–444, July 1997.

[38] L. Vandendorpe and L. Cuvelier, "Statistical Properties of Coded Interlaced and Progressive Image Sequences," *IEEE Transactions on Image Processing*, vol. 8, pp. 749–761, June 1999.