New Techniques for Streaming MPEG Video over the Internet

Jian Zhou

A dissertation submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

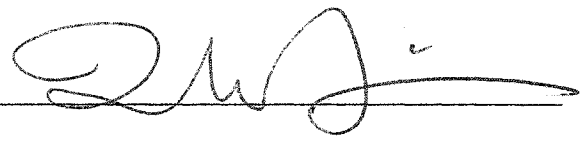
University of Washington

2003

Program Authorized to Offer Degree: Department of Electrical Engineering

Doctoral Dissertation

In presenting this dissertation in partial fulfillment of the requirements for the Doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of the dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to ProQuest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature 

Date 12/04/2003

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Jian Zhou

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of Supervisory Committee:

Ming-Ting Sun

Reading Committee:

Eve Riskin

12/04/03

Radha Poovendran

Ming-Ting Sun

Date: 12/04/03

University of Washington

Abstract

New Techniques for Streaming MPEG Video over the Internet

Jian Zhou

Chair of the Supervisory Committee

Professor Ming-Ting Sun
Department of Electrical Engineering

Many video coding standards have been established for various streaming video applications, such as H.263 for low-bit-rate two-way video communications, MPEG-2 for broadcasting and general high-quality video applications, and MPEG-4 for streaming video and interactive multimedia applications. Since its introduction a decade ago, streaming video over the Internet has been experiencing dramatic growth with the rapid increase of network bandwidth and computing power. The target of the streaming video service is to provide satisfactory visual quality and flexible functionality to the clients. Many issues should be taken into consideration, for example, how to deliver pre-encoded video streams over networks with varying bandwidth, how to improve the error resilience of the video transport over lossy channels, and how to support friendly user-interface (such as providing full VCR functionalities for controlling the playback of the video streams) at the client side.

In this dissertation, we propose several new techniques to address the above issues. In the first part, we present an enhancement layer truncation scheme for the

delivery of the MPEG-4 Fine-Granularity-Scalability (FGS) video. Our target is to reduce the quality variation of different parts within each frame when the last transmitted enhancement layer is truncated according to the available network bandwidth. In our approach, the bit-budget for the truncated enhancement layer is redistributed to each block, so that the whole frame area can be covered uniformly. An operational rate-distortion optimization scheme is further adopted to improve the decoded visual quality and to reduce the intra-frame quality variation. In the second part, we point out that multi-path feature exists in today's networks, and we can make use of it to overcome the problems of instantaneous network congestion and large bandwidth requirement for higher enhancement layers by transmitting the split FGS enhancement layers. We initially split the original enhancement layers evenly into multiple descriptions. Rate-distortion optimization is then carried out on each description to improve the performance. In the third part, we present our solution to provide full VCR functionality for the MPEG video streaming application by adding a reverse-encoded bit-stream and using a minimum-cost frame-selection scheme to minimize the number of frames to be sent over the network and to be decoded. The drift-compensation issue is also taken into consideration. With the proposed scheme, an MPEG video streaming system with full VCR functionality can be implemented to minimize the required network bandwidth and decoder complexity.

TABLE OF CONTENTS

List of Figures

Figure Number Page

List of Tables

Table Number Page

Glossary

DCT: discrete cosine transform

MSE: mean squared error

PSNR: peak signal-to-noise ratio

QP: quantization parameter

SAD: sum of absolute difference

VLC: variable length coding

FGS: Fine Granularity Scalability

QoS: Quality of Service

IQV: Intra-Frame Quality Variation

VCR: Video Cassette Recorder

Acknowledgements

I would like to show my most sincere appreciation to Bing, my dear wife, for her patience, understanding and support.

Thanks to my parents, for their tremendous support since the first day I arrived in this world.

Thanks to Professor Ming-Ting Sun, my advisor, both within and beyond the academic scope, for his invaluable guidance, consistent encouragement and support throughout the years.

Thanks to Professor Eve Riskin and Professor Radha Poovendran, for their inspiring suggestions.

Best wishes to many fellow students in the Information Processing Lab, Jeongnam Youn, Supavadee Aramvith, Jun Xin, Renjit Thomas, Tao Yang, Daniel Gatica Perez, Hsu-Feng Hsiao, Qiang Liu, Yeping Su, Jinhui Pan, Zhi Zhou and Xiaodan Song, for various kinds of help they offered me.

Chapter 1. Introduction

Multimedia applications have entered an exciting era that will enormously impact our daily life. With the development of high performance computing technology, compression technology, and high-speed networks, it is now feasible to provide real-time multimedia services over digital networks, including the Internet and wireless networks. The transport of both live encoded video and pre-encoded and stored video, is an important part of real-time multimedia services. Realizing that video transport is important to multimedia applications, many companies, organizations, and universities are developing products [1-5], standards, and new technologies [6] in this area.

There are three major video transport applications [7]:

- *Complete download,* where the whole video stream is transmitted with a reliable network protocols like ftp [8] or http [9]. This is actually not a real-time application, and the video encoder will not care about the computational complexity, the end-to-end delay, or the characteristics of the delivery channel. However, due to the large volume of the video data, complete download of a video stream may take a very long time and often is not practical for the application.
- *Conversational applications, such as video conferencing*. Such applications are characterized by very strict delay constraints and require real-time video codecs. Due to the strict delay constraints, it is relatively difficult to combat network impairments. For example, retransmission of lost packets and FEC

(Forward Error Correction) with interleaving may not be practical due to the long delay they may incur.

- *Streaming applications, such as distance learning and video on demand [10][11].* This kind of applications allows the users to start playing the video before the whole video bit-stream is received, with an initial delay of only a few seconds. The video streams may need to be adapted to a heterogeneous network environment where different users may have different network bandwidth capacity, varying network impairment characteristics, and user-terminal capability. Video quality, robustness to network impairments, and friendly user control of video playback are important issues for these applications.

For today's Internet structure, it was originally designed for data communication where the best-effort service is offered. Excessive traffic can cause congestions, and video packets may be lost, duplicated, or re-ordered on their way from the source to the destination. Thus extra delay may be introduced, and the received video quality may be degraded. In this environment, no QoS (Quality of Service) is guaranteed for the video transport applications mentioned above in terms of bandwidth, delay, and packet loss [6].

Two approaches have been proposed to address these technical issues [6]. One is *network centric*, where the modification of the current network infrastructure is required to provide the QoS for the multimedia streaming [13][14]. The other one *is end system based*, which does not rely on the modification of the network. Certain Strategies should

be taken on the end system to adapt to the network behaviors, e.g., using error-resilience and error- concealment techniques to cope with the packet losses [12].

In this dissertation, we will concentrate on the end system based approaches since they can be readily implemented without waiting for the large-scale end-to-end modification of the network. Especially, we will investigate techniques related to video encoding - the key technology that enables the streaming video applications.

For video streaming applications, the video has to be compressed. Currently, there are several established video-compression standards established, including MPEG-1/2/4 [28][29][30] and H.261/H.263/H.264 [31][32][33] targeted for different video applications (e.g. H.263 for low-bit-rate video applications, MPEG-2 for high bit-rate high-quality applications). Besides developing compression standards, the MPEG committee, ITU-T standards committee, and many researchers have also been developing various technologies for applications related to the streaming media.

Examples of important techniques for streaming video applications include simulcast, transcoding, multiple description coding, and scalable video coding.

- Simulcast is a technique that enables a video encoder to generate multiple video streams for the same video content at different rates for different network capacities. Each stream can be delivered to the receivers via broadcast or multicast channels [15]. Figure 1 below illustrates this concept. When the network bandwidth matches the bit-rate of an encoded video stream, the optimal quality can be achieved. If the network bandwidth does

not match the bit-rate of an encoded video stream, then an encoded video stream with the bit-rate lower than the network bandwidth has to be used. In this case, the network capacity is not fully utilized and the video quality is not as good as that can be achieved if the network bandwidth could be fully utilized [16]. Another problem with simulcast is how to determine the number of the generated bit streams; if the number is too small, it will not be able to cover the possible bit-rate range, and the provided quality is low if the network bandwidth does not match the encoded bit-rates. If the number is large, the generated bit-streams will occupy too much storage space in the server.

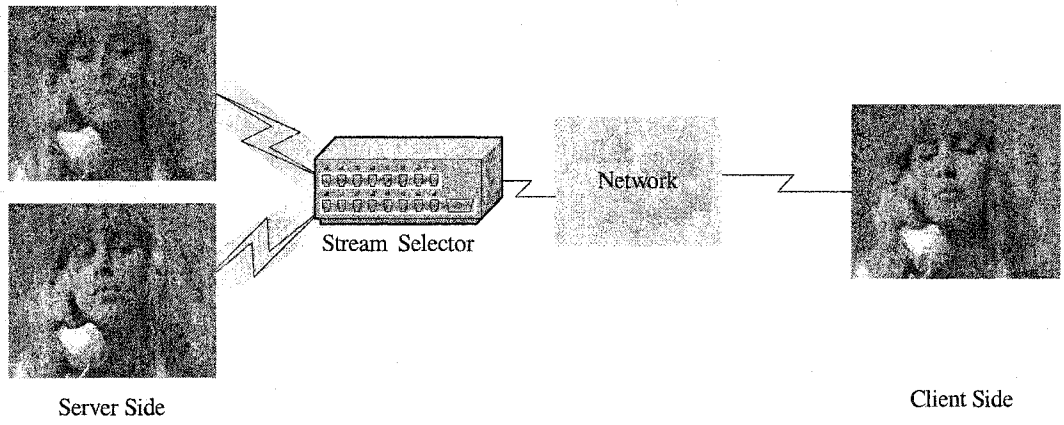


Figure 1. Illustration of the Simulcast Concept

- Video transcoding is the operation of converting a video from a compressed format into another compressed format [17][18]. One of the earliest applications of transcoding is to convert a compressed video, which is originally encoded at a high bit-rate, into a lower bit-rate [19] for

transporting over a lower bandwidth network. This concept is illustrated in Figure 2. A real-time transcoder can adapt the bit-rate in real-time to match the instantaneous network bandwidth as the bandwidth of the network varies. Besides the bit-rate adaptation, a transcoder can dynamically change any coding parameters (e.g., frame-rate and spatial resolution) and/or coding standards (e.g., MPEG-2 to MPEG-4) of the compressed video.

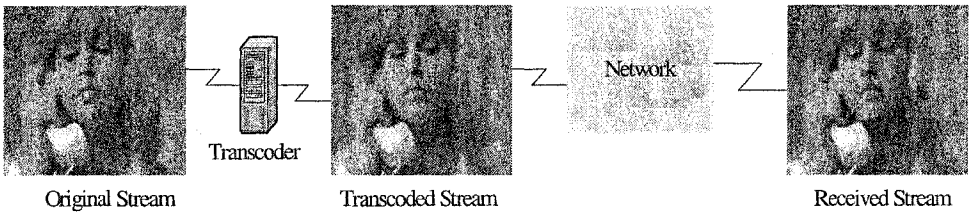


Figure 2. Illustration of the bit-rate transcoding concept

With the capability of dynamically altering the coding parameters of the compressed video, it is expected that video transcoding will play an important role for universal multimedia access by the Internet users with different access links and devices [20]. Envisioning the potential of transcoding, the emerging MPEG-7 standard [21], which standardizes a framework for describing audio-visual content, has defined "transcoding hints" to facilitate the transcoding of compressed video contents [22]. However, the transcoding process will impose extra computational complexity on the server, the intermediate node in the transmission path

(e.g., a gateway), or the client side. This will restrict it from being applied in the application like large scaled video-on-demand services.

- Multiple Description Coding (MDC) addresses the problem of encoding a source into multiple independent bit-streams and transmitting them over a communication system with multiple channels such that a high-quality reconstruction can be achieved by decoding the multiple bit streams together, while a lower quality reconstruction can be achieved if only a smaller number of bit-streams are received due to the network impairment [23]. The concept is illustrated in Figure 3. This technique is appropriate for the packet network when re-transmission is not possible and long delays are not acceptable [24]. More background knowledge and about MDC can be obtained in [24].

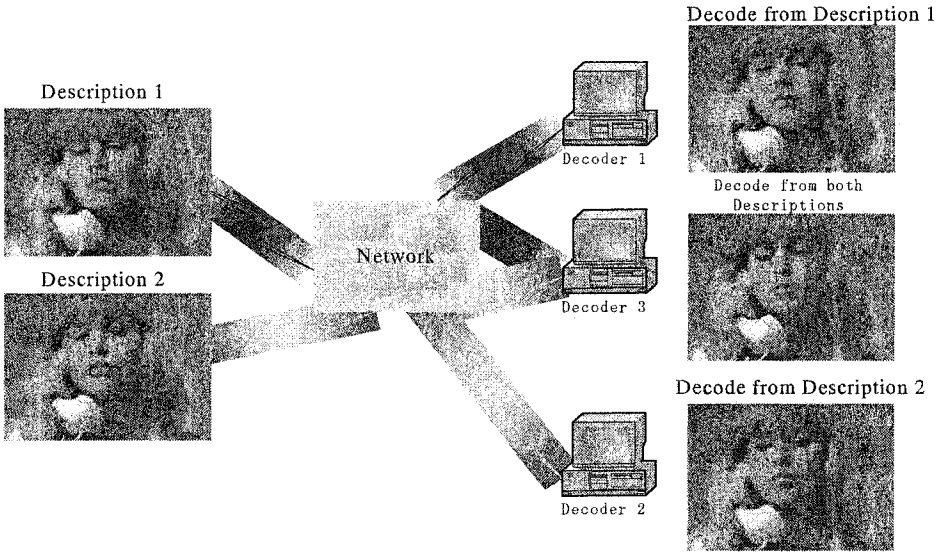


Figure 3. Illustration of the MDC concept

- Scalable video coding (layered coding) encodes the original video content with a base layer and a few enhancement layers. The enhancement layers improve the spatial, temporal, and/or SNR (Signal to Noise Ratio) quality to the reconstructed base layer. More recently, a new form of scalability, known as fine granularity scalability (FGS), has been developed and adopted by the MPEG-4 Visual standard [25]. In contrast to conventional scalable coding schemes, FGS allows for a much finer scaling of the bit-rate in the enhancement layer [16] by a bit-plane coding method of DCT coefficients in the enhancement layer coding [26]. The bit-plane coding of the DCT coefficients allows the enhancement layer bit-stream to be truncated at any point. In this way, the quality of the reconstructed frames is roughly proportional to the number of enhancement bits received.

  Compared to transcoding, scalable coding allows the server to simply truncate the bit-stream to fit the network bandwidth, while transcoding converts the existing data format to meet the current transmission requirements. Scalable coding can provide low-cost flexibility to meet the target bit-rate, spatial resolution, and temporal resolution, by sacrifice the coding efficiency compared to single-layer coding.

  Compared to Multiple Description Coding, layered coding addresses the problems of heterogeneous client bandwidths and dynamic network congestions by means of sequences of layers, while MDC addresses the problem of unreliable channels by means of independent descriptions [27].

The major disadvantage of scalable coding schemes compared to non-scalable coding schemes is the loss of coding efficiency. One important reason of the loss of the coding efficiency is that the base-layer has lower quality and it leads to poor prediction.

In this dissertation, we will discuss new techniques for streaming MPEG video over the Internet and wireless networks. Our approaches are standards-conforming and make use of the flexibility provided by the MPEG-4 FGS properties.

When an MPEG-4 FGS stream is delivered over the Internet or wireless networks, the enhancement layer should be truncated to meet the network bandwidth requirement. Current truncation schemes [42-46] are not able to enhance the whole frame uniformly, which leads to the intra-frame quality variation. In the first part of this dissertation, we present our MPEG-4 FGS enhancement layer truncation scheme. Our target is to minimize the quality variation of different parts within each frame when the last transmitted enhancement layer is truncated according to the available network bandwidth. We first redistribute the bits in the enhancement layer that can only be partially transmitted so that it is able to cover the whole frame area to raise the quality of different parts in the frame uniformly. An operational rate-distortion optimization scheme based on the Lagrange Multiplier (LM) algorithm is further adopted to improve the visual quality and reduce the intra-frame quality variation. Compared to straightforward truncation, our approach reduces the intra-frame quality variation significantly, and the decoded visual quality in terms of PSNR is also improved.

When we transport the FGS streams over lossy channels, with both random errors and burst errors, we need to consider how to improve the robustness of the FGS bit-stream delivery. In the second part of this dissertation, we study the characteristic of the coded FGS enhancement layer, and propose to utilize the multi-path features of today's networks to transmit the split version of the high FGS enhancement layers to overcome the problems of instantaneous network congestion and large bandwidth requirement for higher enhancement layers. We also propose techniques to split the high enhancement layers to reduce their rates for the multi-path transport. We initially split the original enhancement layer blocks by evenly allocating the bits in every block into different channels. Rate-distortion and other optimization schemes are then carried out on each description, so that the split coding efficiency and its robustness can be improved. Compared to the single path transport approach, we can achieve better decoded visual quality.

Another interesting issue related to the streaming video application is to provide full VCR functionality for the MPEG video streaming application. Due to the I-B-P structure of an MPEG encoded video stream, how to efficiently realize the functions of random-access, fast-forward, and fast-backward play of the video in the compressed domain is not a trivial task. In the third part of this dissertation, we propose a solution to this problem. We propose to add a reverse-encoded bit-stream and use a minimum-cost frame-selection scheme to minimize the number of frames and video data to be sent over the network and to be decoded. The drift-compensation issue is also taken into consideration. With this proposed scheme, an MPEG video streaming system with full

VCR functionality can be implemented to minimize the required network bandwidth and decoder complexity.

This dissertation is organized as follows. In Chapter 2, we give a brief overview of video coding and background information related to the issues which will be addressed in this dissertation. In Chapter 3, we discuss the FGS enhancement layer truncation scheme to minimize the intra-frame quality variation. In Chapter 4, we present the proposed multi-path transport of the FGS video. In Chapter 5, we present the proposed scheme to support the full VCR functionality for the MPEG video streaming. Concluding remarks are given in Chapter 6.

Chapter 2. Overview of Video Coding Techniques

2.1. Standard Video Encoders

There are currently several established video coding standards, including MPEG-1 [28], MPEG-2 [29], MPEG-4 [30], H.261 [31], and H.263 [32]. They are all based on the same framework: hybrid DCT (Discrete Cosine Transform) [34] and MCP (Motion Compensated Prediction) coding. Video sequences usually contain redundancy in both the temporal and the spatial dimensions. To reduce the spatial redundancy between nearby pixels within the same image, the DCT is applied to image blocks of 8x8 pixels. Inter-frame DPCM (Differential Pulse Code Modulation) coding techniques employing temporal prediction (motion compensated prediction between frames) are used to reduce the temporal redundancy. In current video coding standards, an adaptive combination of temporal motion-compensated prediction followed by the DCT coding of the remaining spatial information is used to achieve high compression.

In this section, we first describe the generic hybrid MCP/DCT coding framework in the context of MPEG-4 video coding standard.

The MPEG-4 video coding algorithm operates on images represented in the YUV color space. If an image is stored in the RGB format, it should first be converted to the YUV format. In the YUV format, images are also represented in 24 bits per pixel (8 pixels for the luminance signal (Y) and 8 bits each for the two chrominance signals (U and V)). The chrominance signals can be sub-sampled 2:1 in both the horizontal and vertical directions. This sub-sampling does not affect visual quality much because the eye

is more sensitive to luminance than to chrominance signals. Sub-sampling is a lossy step. The 24 bits RGB information is reduced to 12 bits YUV information, which gives a 2:1 compression. The sub-sampled format is called 4:2:0.

Each video frame is divided into 16x16 pixel macroblocks. Each macroblock consists of four 8x8 luminance blocks and two 8x8 chrominance blocks (one U block and one V block). The macroblock is the unit for motion-compensated prediction. Blocks (with 8x8 pixels) are used for DCT compression. Frames can be coded in three types: intra-frames (I-frames), forward predicted frames (P-frames), and bi-directional predicted frames (B-frames).

An I-frame is encoded as a single image, without referencing to any other frames. Each 8x8 block is first transformed from the spatial domain into the frequency domain. Most energy of the block is compacted into the DCT coefficients in the upper-left corner of the resulting 8x8 block in the frequency domain. After this, the DCT coefficients are quantized. Quantization is the only lossy part of the whole compression process other than the sub-sampling. The resulting quantized DCT coefficients are then entropy coded: first run-length encoded in a zig-zag scan order, then variable-length coded. The I-frame in the encoder needs to be reconstructed and stored in the frame memory for future reference. In the reconstruction, the quantized DCT coefficients are inverse quantized and inversed DCT (I-DCT) transformed. Since an I-frame is encoded without referencing other frames, it has the lowest coding efficiency and consumes more bits than P and B frames. However, an I-frame can serve as a random access point for the decoding, in that it can be decoded independently of other frames.

A P-frame is encoded relative to the past reference frame. The reference frame can be a P-frame or an I-frame. The past reference frame is the latest preceding reference frame. Each macroblock in a P-frame can be encoded either as an I-macroblock or as a P-macroblock. An I-macroblock is encoded just like a macroblock in an I-frame. A P-macroblock is encoded as the difference between the current macroblock and a 16x16 area of the past reference frame. To specify which 16x16 area in the past reference frame is used for prediction, a motion vector is included. A motion vector (0,0) means that the 16x16 area is in the same position in the reference frame as that of the macroblock currently being encoded. Other motion vectors are relative to that position. Figure 4 illustrates the concept of motion estimation and motion vector. In MPEG-4 coding, N is 16, and "p" represents the search range. The motion vector is the displacement of the current macroblock with respect to its "best" match block of the same size in the reference frame. Motion vectors may have half-pixel or quarter-pixel precision, where the missing pixels needed for prediction are interpolated. The motion vectors can point to a position outside the boundary of the reference frame, which is called the "unrestricted motion vector" mode. The boundary pixel values are used for those pixel positions outside the frame. The error terms after the prediction are encoded using the DCT, quantization, run-length encoding, and variable-length coding, similar to the I-frame. A macroblock may also be skipped when it results in a (0,0) motion vector and all-zero prediction error terms. The search for a good motion-vector (one that gives small error terms and a good compression) is the most computationally intensive part of an MPEG video encoder and critically affects the resulting video quality. P frames also need to be

reconstructed and stored in the encoder to be the reference frames for (later or previous) P and B frames.

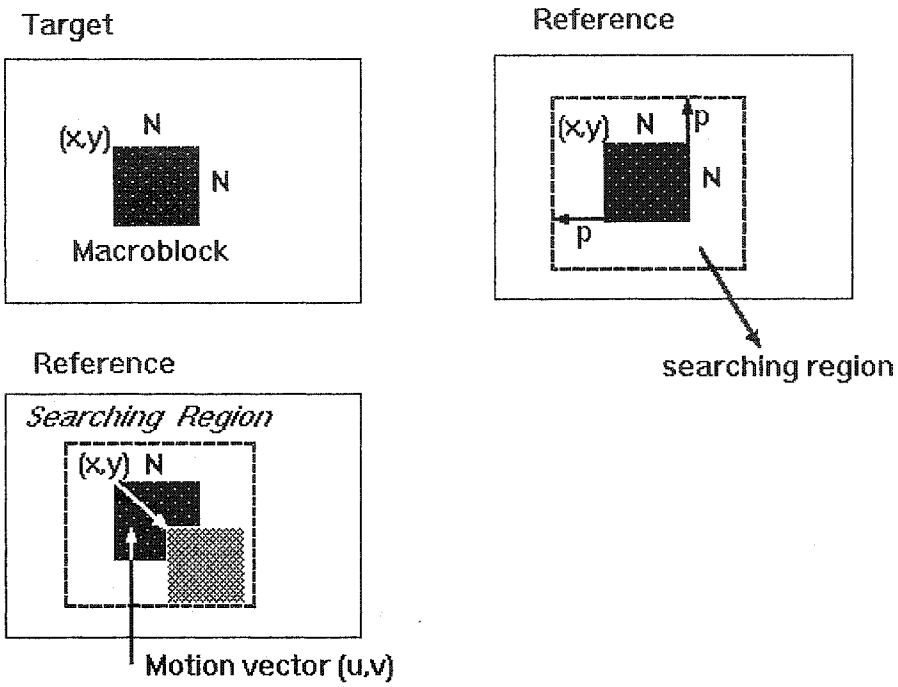


Figure 4. Illustration of motion estimation.

A B-frame is encoded relative to a past reference frame, a future reference frame, or both of them. The encoding for B-frames is similar to P-frames, except that motion vectors may refer to areas in the future reference frames. For macroblocks that use both past and future reference frames, the two prediction areas are averaged before being used for the prediction of the current macroblock. Since a B-frame can take advantage of the bi-directional motion estimation, it has the best coding efficiency among the three frame coding types. However, the bi-directional motion estimation requires a future reference frame, which introduces extra coding delay, and is not suitable for real-time two-way video communication applications.

A typical coded video sequence with all three types of frames is shown in Figure 5. The arrows represent the inter-frame dependencies. Frames do not need to follow a static IPB pattern as shown in the figure. Each individual frame can be of any type. In this dissertation, a fixed IPB sequence is assumed throughout the entire video stream for simplicity. An input sequence encoded with an IBBPBBP structure will produce an output sequence with the structure IPBBPBB.

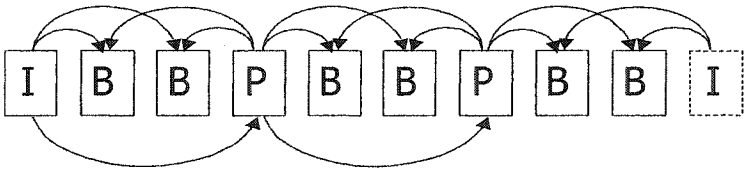


Figure 5. Example MPEG video frame-type mixing pattern.

A block diagram of a typical hybrid MCP/DCT video codec is shown in Figure 6. The encoder generates a variable bit-rate bit-stream. Therefore, an encoder-buffer is needed to smooth out the bit-rate so that the averaged output bit-rate matches the channel bit-rate. To prevent the encoder-buffer from overflow or underflow, and to achieve the best overall video quality, a rate-control scheme is applied to adjust the quantization step-size so that consistent video quality can be obtained. The decoding is simply a reverse process of the encoding.

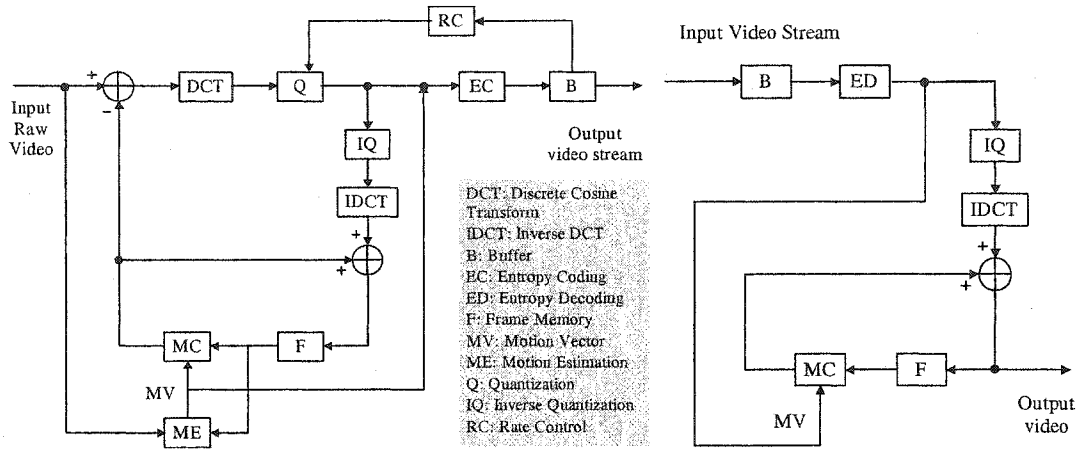


Figure 6. A hybrid MCP/DCT video codec.

A compressed MPEG-4 video sequence is an ordered bit-stream conforming to the standard syntax. Each video sequence is composed of a series of Groups of Pictures (GOP's). A GOP is composed of a sequence of pictures (frames) beginning with an I-picture. A frame is composed of a series of slices. Each slice can provide a slice synchronization code, which can help the recovery after a transmission error corrupted the bit-stream. A slice is composed of a series of macroblocks, and a macroblock is composed of 6 blocks (four for luminance and two for chrominance). The GOP structure is intended to assist random access into a sequence. A GOP is an independently decodable unit that can be of any size as long as it begins with an I-frame. A GOP is independently decodable if it is closed. For example, a GOP with the pattern IBBP is closed, but a GOP with the pattern IB is not, since the decoding of the B-picture requires the reference picture in the next GOP.

2.2. Overview of Major Video Coding Standards

H.261 was developed for videoconferencing using ISDN channels. The major characteristics are:

- H.261 only supports I- and P- frames due to the low-delay requirement.
- H.261 does not support half-pixel motion compensation.
- An optional loop filter [35] may be used to low-pass filter the reference frame, which usually decreases the blocking effect of the reference frame.

H.263 is an improved version of H.261, aiming at low-bit-rate (< 64 kbps) video communications. The major improvements over H.261 include:

- Half-pixel motion compensation.
- Improved VLC coding.

In addition to these improvements, H.263 offers a list of optional features as annexes, including unrestricted motion vectors, advanced prediction mode, arithmetic coding, and deblocking filter. In order to enable the transport of H.263 video over unreliable networks, a set of tools have been developed for the purpose of error resilience, and have been included as additional annexes.

MPEG-2 targets to produce TV-quality pictures at data-rates of about 4-8 Mbps and transparent quality pictures at about 10-15 Mbps. MPEG-2 deals with high-quality coding of possibly interlaced video (e.g. SDTV or HDTV). A wide range of applications is addressed, including all forms of digital storage media, television broadcasting, and communications. The major features of MPEG-2 are:

- MPEG-2 supports interlaced video sequences, and as a consequence, MPEG-2 allows additional scan patterns for DCT coefficients and motion compensation with blocks of 16x8 pixels.
- Additional techniques to improve the video coding efficiency, including ten-bit quantization for DC coefficients, non-linear quantization, and better VLC tables.
- MPEG-2 supports various modes of scalability, including spatial, temporal, and SNR scalability.
- MPEG-2 introduces the concept of Profiles and Levels, where a *profile* defines a set of coding tools or algorithms that can be used in generating conforming bit-streams for some applications, and a *level* places constraints on some key parameters in the bit-streams of a profile.

MPEG-4 is designed to address the requirement of the next generation of interactive multimedia applications, as well as traditional multimedia applications. The concept of Video Object (VO) is proposed, corresponding to the entities in the bit-stream that a user can access and manipulate. An instance of VO's at a given time is called a Video Object Plane (VOP). When the sequence has only one rectangular VOP of fixed size displayed at a fixed interval, it corresponds to the frame-based coding technique, which is of our primary interest. MPEG-4 video defines several profiles to address different applications. The major profiles are:

- Simple Profile (SP): It is based upon the baseline H.263. MPEG-4 SP is targeted at low-bit-rate, low-delay video communications. It includes a set of error resilience tools.
- Streaming Profile: It uses the Fine Granularity Scalability (FGS) coding technique. An FGS video stream can be truncated at any point in its enhancement layers, and is particularly suitable for applications involving streaming video over non-guaranteed Quality of Service (QoS) networks. FGS is described in more detailed in the next section.
- Advanced Simple Profile (ASP): It includes advanced coding tools such as Global Motion Compensation (GMC), quarter-pixel motion compensation, bi-directional Video Object Plane (B-VOP), and interlaced video, etc, to provide better coding efficiency for low-bit-rate video communications.
- **MPEG-4 AVC** (Advanced Video Coding)**/H.264 [33]**: It is the latest video coding standard and was just completed in the year 2003. It reflects the latest advances of video coding techniques. The major improvements over H.263 and previous MPEG-4 video coding profiles are: supporting multiple reference frames, variable block-sizes for motion compensation (up to 7 different block-sizes), 4x4 integer DCT-like transform, improved intra prediction, context adaptive binary arithmetic coding, etc. [36]

2.3. Scalable Video Encoders

The target of the traditional single layer video encoding is to optimize the video quality at a fixed bit-rate [16]. However, for streaming applications, the network is often a heterogeneous environment of IP, ATM, mobile networks, etc, and different users may have different access to it. LAN access bandwidth usually is of mega-bits/s. DSL and Cable provides access bandwidth from several hundred-kilo bits/s to a couple of mega-bits/s. Traditional telephone modem users have a bandwidth range from 14.4 to 56 kbits/s. The access bandwidth for cellular phones currently is only a few kilo bits/s. Besides, these links have different channel characteristics, and the bandwidth may vary drastically. Also, the device to access the networks may have very different capability. At present, personal computers are the major Internet access devices, while devices including handheld computers, personal digital assistants (PDA's), set-top boxes, screen telephones, smart cellular phones, and network computers, are expected to become the dominant access terminals for accessing the Internet [10]. These network terminals (including PCs and network appliances) vary a lot in computing power and display capability. In addition, users' interests in the content may differ from each other [37]. As a result, for the compressed video to be intelligently delivered to users with different available resources, access links, and interests, the content must be adapted dynamically according to the network condition and the different requirements of different users.

The idea of "scalable coding" is one approach proposed for the streaming video applications with varying spatial-temporal and quality fidelities. In scalable coding schemes, the original video content is encoded into several layers. One of them is the

base-layer to provide the basic visual quality, and the other layers are called the enhancement-layers to further enhance the decoded visual quality. The complete bit-stream (i.e., the combination of all the layers) provides the highest quality. Decoding only the base-layer or the partial enhancement-layers produces pictures with degraded quality, lower image resolution, or lower frame-rate. The scalabilities of quality, image resolutions, or frame-rates, are called SNR, spatial, or temporal scalability, respectively. These three scalabilities are basic scalable mechanisms. They can be combined together to offer spatial-temporal scalability [38][39].

Signal-to-noise ratio (SNR) scalability is a scheme to encode a video sequence into two layers with the same frame rate and the same spatial resolution, but with different quantization accuracy. The base-layer is generated by a single-layer MPEG encoder with coarse quantization parameters, while the enhancement layers are generated by quantizing the residual errors between the original pixel values and the reconstructed base layer pixel values in the DCT domain. The quantization parameter used for the enhancement layer is smaller than that adopted by the base-layer encoder. The diagram of a SNR scalable video encoder is depicted in Figure 7 below.

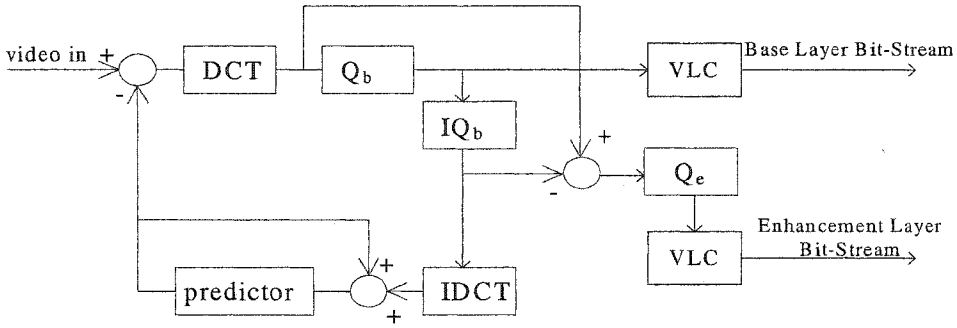


Figure 7. Diagram of a SNR scalable video encoder

Spatial scalability is a scheme to encode a video sequence into two layers at the same frame rate, but different spatial resolutions. The base layer is coded at a lower spatial resolution. The reconstructed base-layer picture is up-sampled to form the prediction for the high-resolution picture in the enhancement layer. A block diagram of two-layer spatial scalable encoder is shown in Figure 8 below.

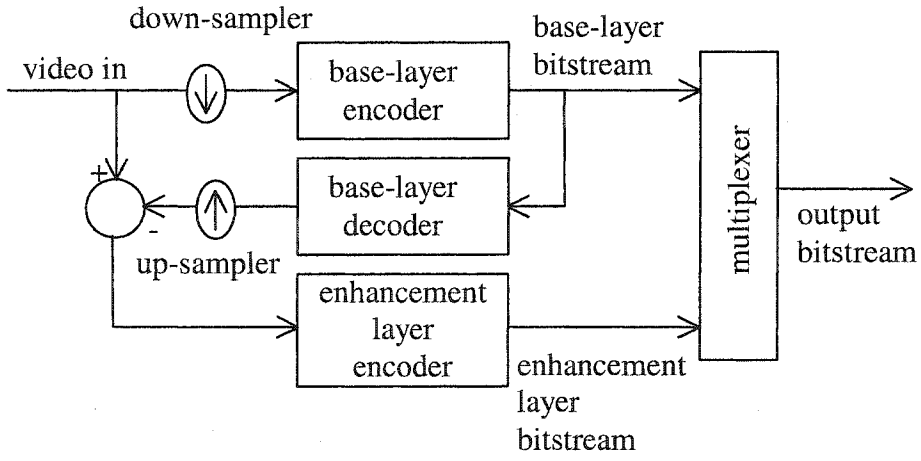


Figure 8. A block diagram of two-layer spatial scalable encoder

Temporal scalability is a scheme to encode a video sequence into two layers at the same spatial resolution, but different frame rates. The base layer is with a lower frame rate. The enhancement layer provides the missing frames to form a video with a higher frame rate. Actually, the B frame in the MPEG video sequence can be dropped for the temporal scalability purpose.

As described by Li [16], for the traditional scalability, a common characteristic is that the enhancement layer should be entirely transmitted, received, and decoded, or it does not provide any enhancement at all.

In order to provide more flexibility in meeting different demands of video streaming, a new scalable coding mechanism, called Fine Granularity Scalability (FGS), was proposed to MPEG-4. As shown in Figure 9, the FGS encoder compresses a raw video sequence into two streams, a base layer bit-stream and an enhancement layer bit-stream. The base layer encoder structure is the same as what we have described in Section 2.1, a generic hybrid MCP/DCT video encoder. Different from an SNR-scalable encoder, an FGS encoder uses bit-plane coding to represent the enhancement stream (see Figure 10). The following example is taken from [16] and illustrates the bit plane coding procedure.

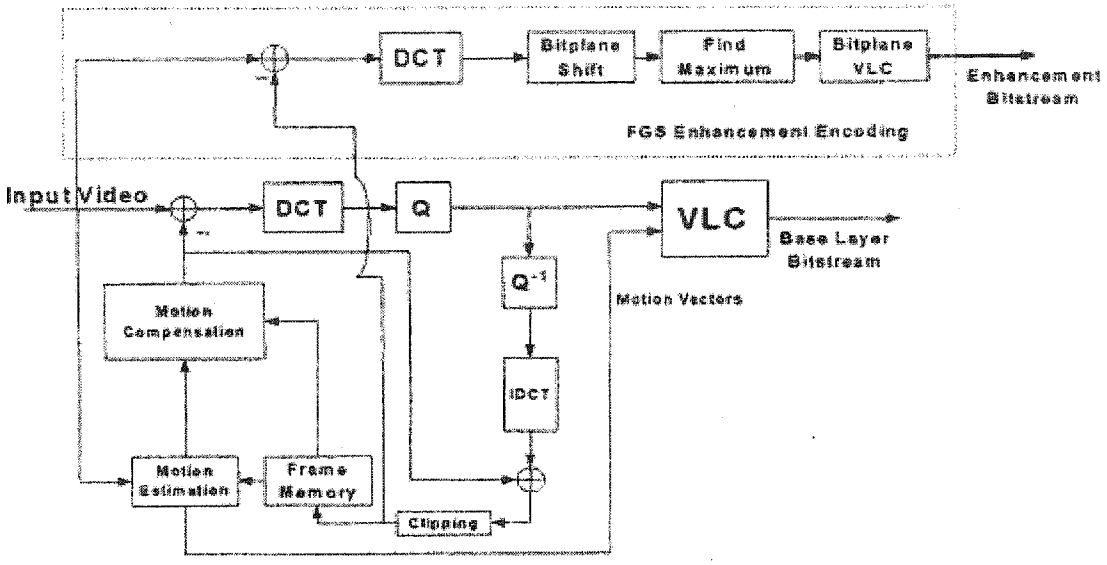


Figure 9. MPEG-4 FGS Encoder [16].

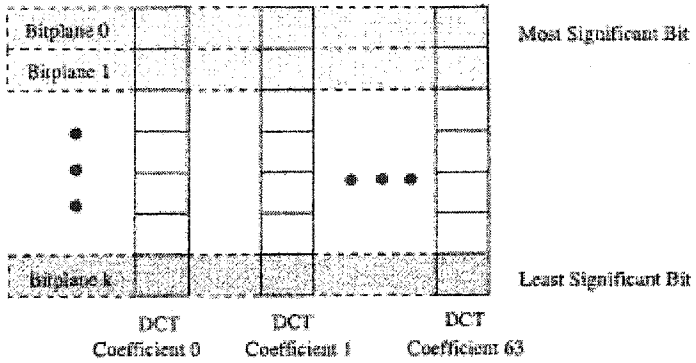


Figure 10. Bit planes of enhancement DCT coefficients[12]

Assume that the absolute values and the sign-bits after zigzag ordering are given as follows:

10 0 6 0 0 3 0 2 2 0 0 2 0 0 1 0 … 0 0

The maximum value in this block is found to be 10 and the number of bits to represent 10 in the binary format (1010) is 4. Therefore, 4 bit-planes are considered in

forming the (RUN, EOP) symbols. Writing every value in the binary format, the 4 bit-planes are formed as follows:

MSB:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	...	0	0
MSB1:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	...	0	0
MSB2:	1	0	1	0	0	1	0	1	1	0	0	1	0	0	0	0	...	0	0
MSB3:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	...	0	0

Converting the four bit-planes into (RUN, EOP) symbols, we have

(0,1)						MSB
(2,1)						MSB-1
(0,0)	(1,0)	(2,0)	(1,0)	(0,0)	(2,1)	MSB-2
(5,0)	(8,1)					MSB-3

Therefore, 10 (RUN, EOP) symbols are formed in this example. These symbols are coded using variable-length codes together with the sign-bits. Each sign-bit is put into the bit stream only once right after the VLC code that contains the MSB of the nonzero absolute value associated with the sign-bit. For example, no sign-bit follows the second VLC code of the MSB-2 plane because the sign-bit has been coded after the VLC code in the MSB-1 plane.

With bit-plane coding, an FGS encoder is capable of achieving continuous rate control for the enhancement stream. This is because the enhancement bit-stream can be truncated anywhere to achieve the target bit-rate.

Many research activities have been conducted in the FGS area. It has been reported that compared to the single layer coding at the same bit-rate, FGS may lose a coding gain of up to 2 dB. In order to improve its coding efficiency, Kalluri [40] and Wu [41] propose to reconstruct partial enhancement layers, and add them back to the reconstructed base layer, so that the quality of the reference for the base layer motion

compensation can be improved, thus leading to a better coding gain. Another area of FGS related research activities is on how to truncate the enhancement layer during the video transport time. Zhao [42][44], Zhang [43] [54], Cheng [55], and Wang [53] propose different schemes aiming at minimizing the quality variation between the adjacent frames, while Cheong [45], Lim [46], and Zhou [48] focus on the reduction of the intra-frame quality variation. How to modify the FGS coding structure to add temporal [56] and spatial [57] scalability, and how to protect an FGS stream during the transmission have also been studied. The techniques described in [49] and [50] can be used for the base-layer coding and transmission. Wang [51] proposes to use the adaptive FEC to protect the FGS enhancement layers. Schaar [52] uses unequal error protection to transport the FGS bit streams. P. Chou [27] focuses on the multicast scenario of the FGS video, while Zhou [47] also proposes some schemes by taking advantage of the multi-path transport features.

2.4. Summary

In this chapter, we reviewed several video coding standards and techniques.

In Section 2.1, we gave an overview of the coding framework that all modern video coding standards are based upon: the hybrid MCP/DCT coding. Then, major video coding standards were briefly described in Section 2.2.

In Section 2.3, we reviewed the scalable video coding technique, especially the MPEG-4 Fine Granularity Scalability. Major research activities in this field are also introduced.

Chapter 3. FGS Enhancement Layer Truncation with Reduced Intra-Frame Quality Variation

3.1. Introduction

For today's Internet video streaming applications, one important concern is to flexibly deliver compressed video streams to the end-users with heterogeneous environments. Fine Granularity Scalability (FGS) has been adopted as an amendment to the MPEG-4 standard [25] to address this concern.

The encoder structure of an FGS encoder is illustrated in Figure 9, where two bit streams are generated: one is the base-layer stream to provide the basic visual quality, and the other is the enhancement-layer stream to improve the base-layer quality. The enhancement-layer stream is encoded into several layers with a bit-plane coding [26] scheme, and this enables FGS to provide continuous rate-control, since the enhancement layer can be truncated at any point to achieve the target bit-rate. The corresponding quality of the reconstructed frames is roughly proportional to the amount of enhancement-layer bits received. The rate and quality of the base layer is assumed to be the lower bound of the rate and quality of the application, since the decoder needs to completely receive the base-layer in order to be able to decode the video. Enhancement-layer bits cover the range of bit-rates from this lower bound to near lossless quality. However, the standard does not specify how to truncate the enhancement layers; it only specifies how to decode the truncated bit stream.

Many research efforts have been introduced on how to truncate the FGS enhancement-layer bit-stream to reduce the video quality variation. One framework is proposed in [56], where the available total bit-budget is first guaranteed to transmit the base-layer frames, and the remaining bit-budget is evenly allocated to each enhancement-layer frame. We call this enhancement layer truncation as "even truncation." This scheme has two problems and will not result in uniform video quality for the decoded frames. The decoded visual quality will vary from frame to frame. The reason is that, the base-layer stream needs to be coded in a low bit-rate in order to fit into different network conditions. With the low bit-rate base-layer, there is less room for the encoder to achieve a constant quality from frame to frame. The enhancement layers are used to enhance the base-layer quality. With even-truncation, roughly the same amount of additional quality will be enhanced for each frame, and there will still exist quality variation for different frames with different complexity. We call this kind of quality variation inter-frame quality variation. The quality of different parts in a same frame will also vary from place to place. The current MPEG-4 FGS uses a normal scan order, as shown in Figure 11, to encode the enhancement-layer (EL) macroblocks from the upper-left corner down to the bottom-right corner in a frame. When the enhancement layer is truncated due to the network bandwidth constraint, the last bit-plane of the enhancement layer will usually cover only part of the frame after the truncation. At the decoder side, the upper part covered by the transmitted bit-plane will be enhanced, and the lower part of the frame will not get the enhanced quality, thus the effect, which we call intra-frame quality variation, will arise. One example of intra-frame quality variation is shown in Figure 12.

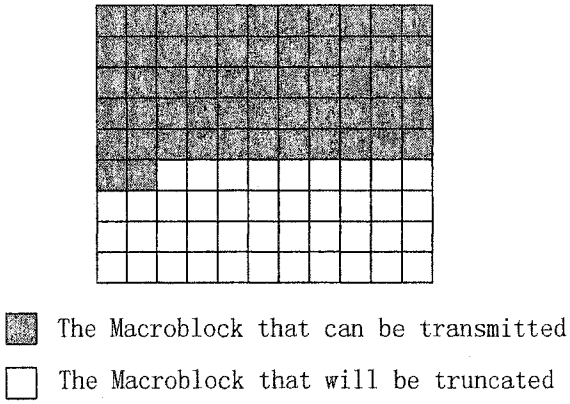


Figure 11. Effects of FGS enhancement layer bit-plane truncation with normal scan order

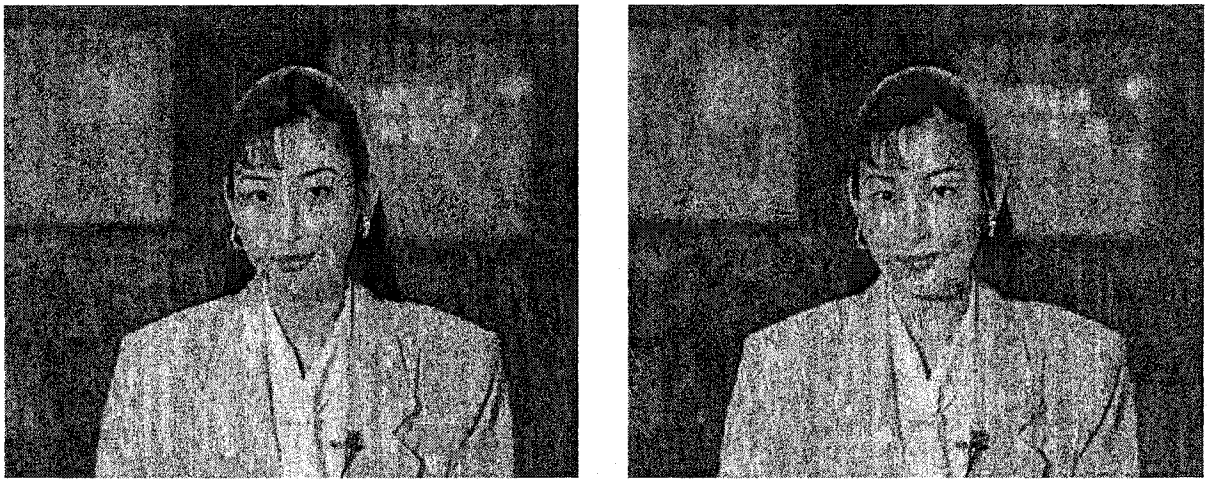


Figure 12. Frame decoded from the whole EL 3 (left) and partial EL 3 (right)

In order to solve the first problem (inter-frame quality variation), Zhao [42] proposes to uses the frame distance from the nearest-feather-line (NFL) [62] to evaluate the importance of each frame, and truncate the enhancement-layers according to the distance and the size of the enhancement layer to be truncated. Zhao [44] and Zhang [43] [54] propose to use the optimal rate-allocation to truncate the enhancement-layer bit-stream and to minimize the sum of the absolute differences between adjacent frames under the rate constraints. The rate-distortion (R-D) curves for each enhancement-layer

frame are interpolated during the encoding time to determine the amount of bits that should be truncated. This algorithm minimizes the inter-frame quality variation, and can be applied to both the single stream truncation and the multiple stream multiplexing. Cheng [55] further develops this scheme by using a composite R-D analysis to minimize the dynamic range of the distortions of the decoded frames. Wang et al. [53] studies the problem of rate-allocation in the enhancement layer for the PFGS (Progressive FGS) coding scheme. An exponential model is used to realize the optimal rate-allocation. Average PSNR improvements in the range of about 0.3-0.5 dB have been reported. However, none of the schemes mentioned above considered the intra-frame quality variation.

For intra-frame quality variation, Cheong [45] uses a water-ring scan order together with selective enhancement features provided by the FGS coding scheme to transmit the bit-planes in the "area of interest" prior to transmitting the bit-planes in other areas. The bit-planes of the area of interest are shifted up, so that when the enhancement bit-stream is truncated, the area of interest may not be affected. However, the decoder needs to be modified to decode the water-ring scanned enhancement layers. Another problem is that, for many video sequences with natural scenes, it is hard to define the area of interest, or there may be more than one area of interest. Lim [46] proposes to re-order the enhancement-layer macroblocks according to the quantization values and the number of the coded DCT coefficients in the corresponding base-layer macroblock. However, similar to the water-ring approach, this method does not solve the problem of intra-frame

quality variation when the enhancement layer is truncated, and the decoder also needs to be modified to decode the enhancement layer.

In this chapter, we propose a standard-compatible truncation scheme to reduce the intra-frame quality variation. Our approach is to re-encode the last enhancement layer that can be transmitted for each frame using the available bit-budget, so that the re-encoded layer can cover the whole frame area and can help to raise the quality of different parts of a frame uniformly. To improve the video quality, a rate-distortion optimization using Lagrange Multiplier is proposed for the enhancement layer truncation. Simulation results confirm the effectiveness of the proposed method.

The rest of this chapter is organized as follows. Section 3.2 presents our simple enhancement-layer truncation approach as well as our rate-optimization approach. Simulation results are shown in Section 3.3, and the summary is presented in Section 4.

3.2. Our Proposed Enhancement Layer Truncation Scheme

3.2.1. Block Bit-Reallocation Truncation Scheme

From the discussion above, we know the reason why the even-truncation led to the intra-frame quality variation is that only a partial frame area can be enhanced. If the last bit-plane to be transmitted can cover the whole frame, the quality of the whole frame can be enhanced uniformly. However, the channel bandwidth is often not wide enough to transmit the whole bit-plane.

We solve the above problem by re-encoding the last bit-plane for each frame. We reduce the bits for each block in the frame in order to distribute the bits for the last

enhancement bit-plane to cover the whole frame area. The reduced bits for each block is proportional to the number of bits generated in each block in the original enhancement layer. Compared to the original last bit-plane, each re-encoded block will have fewer bits than the original one, but the total amount of bits of the last bit-plane will be the same as the original one. The effect is that, the transmitted bit stream is now able to cover the whole frame area, thus the quality of every block will be uniformly enhanced. The whole process to transcode the enhancement layer of one frame is explained below:

1. Encode the current bit-plane as described in the standard, record the amount of bits generated by each block as R_i where $i = 0,1,...N-1$, (N is the number of blocks in the frame) and the total amount of bits R_{BP} for the whole bit-plane.
2. If the bandwidth allows, transmit the current bit-plane and go to step 1 to encode the next bit-plane.
3. If not, it means the remaining bit-budget R_{Budget} will not be enough for the whole bit-plane. The following steps are taken to reduce the number of bits generated in the bit-plane:
   - Shrink the bit budget for each block as:

$$R_i^{'} = R_i - \frac{R_i}{\sum_{i=1}^{N} R_i} \times (R_{BP} - R_{Budget}) \qquad (1)$$

where $R_i^{'}$ is the new bit-budget to encoder each block. Equation 1 indicates the over-shot bit budget $(R_{BP} - R_{Budget})$ is allocated to each block by its original bits contribution to the whole frame.

- Re-encode the symbols in each block from the very beginning until the new bit-budget $R_i^{'}$ is met. In each enhancement layer block, there are 64 bits, either "0" or "1," corresponding to the residual errors from the DC coefficient to the highest AC coefficient. The encoding procedure with the new bit budget means some of the "1" bits used to enhance the high frequency DCT coefficients will be dropped. For a simple scheme, we can just drop the "1" bits in the enhancement layer block from which corresponds to the higher frequency DCT coefficients until the bit-budget is met. A more elaborated scheme using rate-distortion optimization to decide which bits to drop will be discussed in the next sub-section.
- Process the next block until the end of the frame.

Our proposed scheme can be combined with other algorithms aiming at reducing the inter-frame quality variation, such as [54], so that both the inter-frame and the intra-frame quality variation can be reduced. This scheme is fully standard compatible, and it only introduces a small portion of extra processing on the enhancement layer, thus it can be realized in real-time.

3.2.2. Optimization for Enhancement Layer Truncation

In the proposed enhancement layer truncation scheme above, we simply drop the "1" bits in the enhancement layer block from which corresponds to the highest AC frequency in the DCT domain to meet the new bit-budget. Although from simulations, it can reduce the intra-frame quality variation, this scheme is not optimized from the rate-distortion point of view. An example is shown in Table 1. Assume in an enhancement layer block, there are only three coefficients, 8, 0 and 15. They can be represented as "1000," "0000" and "1111" in the binary format. The MSB (Most Significant Bit) bit-plane, or the first enhancement layer can be represented as "101," which contains two "1" bits. Suppose only part of the MSB bit-plane is allowed to be transmitted, we then need to drop some "1" bits in MSB bit-plane. If we decide only to transmit the "1" bit corresponds to coefficient "8," we will need 3 bits to encode the MSB bit-plane according to the Huffman table defined in the FGS standard. We can reconstruct the residue coefficient of "8" at the decoder, but we will lose the coefficient of "15," and the overall distortion will be 225 in terms of the sum of square difference (SSD) for the decoded block. On the other hand, if we decide to keep the "1" bit corresponds to coefficient "15," 5 bits are required to encode this MSB block. We will reconstruct the residue coefficient of "15" as "8" at the decoder side since the lower significant bits are not transmitted, and we will also lose the coefficient of "8" as well. The overall distortion for the decoded block will be as large as 113 in terms of SSD. We can see from this example that dropping different "1" bits will lead to different rate and distortion performance. Some balance should be made to decide which "1" bits in the current block

should be dropped, or kept to achieve the optimal performance in the rate-distortion sense.

Table 1. Example of the Bit Truncation in an MSB Block

	Drop the 2nd "1" bit	Drop the 1st "1" bit
Symbol to be encoded	100 -> (0,1)	001 ->(2,1)
Bit amount	3	5
SSD	15*15=255	8*8+7*7=113

3.2.3. Rate-Distortion Optimization using Lagrange Multiplier

The enhancement layer truncation problem can be generalized as to select some "1" bits from the original block in the last layer, so that the encoded bit-stream conforms to the restricted bit-budget and offer an optimized quality in the rate-distortion sense.

Thus, our optimization target is, given a constraint bit budget R_{Budget} for enhancement layer frame i, find a proper bit pattern for each block to achieve

$$\min \quad D_i \tag{2}$$

subject to

$$R_i < R_{Budget} \tag{3}$$

where R_i is the number of bits to encode the whole frame under certain bit dropping patterns for each block, and D_i is the distortion of the whole frame.

The constrained optimization problem can be solved by dynamic programming [58]. Although it is the optimal solution to the problem, its enormous computational

requirement prevents it from being applied to practical video coding applications. The operational Lagrange Multiplier (LM) algorithm [58][59] approaches the optimal solution with reduced computation. It is proved in [60] that this kind of constrained optimization problem can be converted into an unconstraint optimization problem as:

$$\min \quad (D_i + \lambda R_i) \tag{4}$$

where λ is a positive constant. It is also proved in [60] that the solution for equation (4) can be obtained by minimizing the cost function in each block, i.e.,

$$\sum_{j=1}^{M} \min \quad (d_{ij} + \lambda \times r_{ij}) \tag{5}$$

where r_{ij} is the bits needed to encode the block j in the frame under certain bit dropping patterns, d_{ij} is the associate distortion, and M is the total number of blocks in the frame.

To calculate the cost function in Equation (5), we need to decide the bit drop pattern for every block, and to determine the parameter of λ for the whole frame.

In one enhancement layer block, there are 64 bits in one bit-plane. Each bit can be kept or dropped. The combination of the available drop pattern will be exponential to the number of "1" in the current block. We use the trellis search method, which is illustrated in Figure 13, to find a bit drop pattern under a given value of λ.

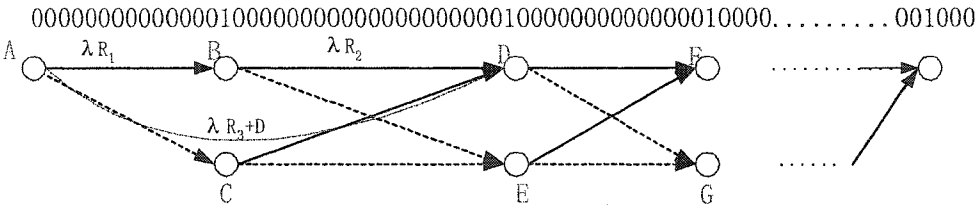


Figure 13. Trellis Search for the Bit Drop Pattern

- A is the starting stage of the bit-plane.
- When it reaches the 1st "1" in the bit plane (we call it a new stage), there are two ways to deal with it: either keep it as "1" or modify it as "0," thus two states are generated, namely, "B" and "C". For the route "A-B," a cost function can be calculated as $J = \lambda R_1$, where R_1 is the length of the code word to describe the bit string so far. For the route "A-C," no cost function is available yet.
- When it reaches the 2nd "1" in the bit-plane, 4 routes are generated, namely, "BD," "CD," "BE," "CE." State "E" indicates that this "1" is modified as "0," and state "D" indicates the "1" is kept. For the two routes entering the state "D," one should be discarded, according to the value of $\lambda(R_1 + R_2)$ (corresponding to the route ABD) and $\lambda R_3 + D$ (corresponding to the route ACD), where R_3 is the length of the code word to describe the bit string of "ACD," and D is the distortion incurred by changing the "1" in position "B" to "0". When computing the distortion occurred by dropping the bit "1" in the current bit-plane,

the bits associated with the same DCT coefficient in the lower significant bits in the enhancement layer should also be taken into consideration.

- The above procedure continues until the end of the block, and one local optimal route will be generated.

To find the λ that results in the optimal solution while satisfy the target-bit constraint, we use a fast convex search algorithm proposed in [61]. The algorithm first finds two boundary λ values λ_1 and λ_2 ($\lambda_1 < \lambda_2$). λ_1 generates more bits than the target bit-number, and λ_2 generates fewer bits than the target bit-number. The optimal λ^* will be between these two values, i.e., ($\lambda_1 < \lambda^* < \lambda_2$). The bi-sectional algorithm can then be used to search for the optimal λ^* [61].

The whole optimization procedure is summarized below:

Step 1 – *Initialize the Lagrange Multiplier* λ. We initialize the λ to the values of the previous picture, i.e., $\lambda = \lambda'$, where λ' is the Lagrange Multiplier of the previous picture.

Step 2 – *With the* λ*, follow the trellis search pattern in Figure 13 to minimizes the cost function* $J=D+\lambda R$ *for each block.* Suppose after encoding the picture, the rate generated for the picture is R. If $0 \le |R_t - R| \le TH$ or a preset maximum number of iteration is reached, stop the search. Otherwise continue to the next step.

Step 3 – *Update the Lagrange Multiplier.* Denote *R1* and *R2* as the rates generated from the current and the previous iterations. $(R1-R_t)(R2-R_t)<0$ means that two

boundary λ's have been found. We use the following pseudo C-code to explain the updating.

```
If ((R1-Rt)(R2-Rt)<0)
{
  /* Use bi-sectional algorithm to update the λ */
  /* (D1, R1) and (D2, R2) are the distortion-rate pairs associated with
     the two boundary λs */
  λ = -(D1-D2)/(R1-R2) ;
}
else
{
  /* Update λ to find the boundary */
  if (R<Rt)
        λ = λ/2;
   else
        λ = λ*2;
}
```

3.2.4. Complexity Analysis of the R-D Optimization Algorithm

Compared to the "even truncation" scheme and the proposed simple re-encoding method in Section 3.1, the proposed R-D optimization algorithm requires some extra

computations, both from the trellis search in the blocks and from the iteration procedure to find the optimal value of λ.

For the trellis search part, the extra computation for each block and the extra space to store the information of the temporary routes are all linear functions with respect to the number of "1" bits in the block. In each stage, only 1 route entering the "1" state will survive with the minimum cost function up to the current stage. The number of the routes entering the state "0" is the sum of one route from the "1" state and other routes from the "0" states, both in the previous stage. From Figure 13, we can figure out that there is only one route from the previous "0" state in stage 1, two routes in stage 2, etc. It can be proved that in the stage n (the n-th "1" bit in the current block, but not the last "1" bit in the current block), there are n such routes to enter the current "0" state. For the last "1" bit in the block, all the temporarily stored routes will converge into one route, which indicates the optimal "1" bits drop pattern. Since in each block, there're only 64 bits, the total number of routes that needs to be saved temporarily during the trellis search is at most 64, which happens when all the bits are "1". However, in practical situations, the number of bits is usually much less than 64. Table 2 below shows the statistical result of the maximum/minimum/average number of "1" bits in one block for some video sequences simulated. With the small number of "1" bits in practical situations, the proposed algorithm does not introduce much extra computations.

Table 2. Maximum / minimum /average number of "1" bits in the blocks of each enhancement layer

Sequence	Enhancement Layer	Max	Min	Average
Football (CIF)	1	8	0	0.12
	2	26	0	2.07
	3	33	0	6.05
	4	41	0	11.30
News (QCIF)	1	7	0	0.07
	2	21	0	1.71
	3	31	0	4.49
	4	41	0	8.04

For the extra computation incurred from the iterations to decide the optimal λ, it depends on the number of iterations needed until the result is converged. In order to minimize the extra computations incurred from this part, we take the following steps:

- For the first frame in the video sequence or in a new scene:

  For the first frame in a new scene, it has different statistical characteristics from its previous frames, and the optimal λ value may be different. In [36], it is concluded that for the single layer video encoder, when rate-distortion optimization is applied, the optimal λ value can be approximated as:

$$\lambda = 0.85 \times QUANT^2 \tag{6}$$

where QUANT is the average quantization parameter of all macro-blocks. This scheme has been applied to many video coding rate-control algorithms. For the FGS enhancement layer encoding, similar idea can be applied by introducing an equivalent quantization parameter. Actually, for the enhancement layer block, when n bit-planes can be sent out, the equivalent quantization parameter can be defined as

$$Q_e = \frac{Q_b}{2^{n-1}} \tag{7}$$

where Q_b is the quantization parameter for the base layer block. Then, we can associate the initial λ value with the equivalent quantization parameter as

$$\lambda = 0.85 \times Q_e^{\,2} \tag{8}$$

We can set the first initial λ value from Equation 8, and follow the steps described in the above section to find two boundary λ values, and then take iterations until the optimal λ value is achieved.

Table 3 below shows the average number of iterations needed to find two boundary λ values, and the number of iterations needed to compute the optimal λ from these two boundary values.

Table 3. The average number of iterations to find the initial λ and the average number of iterations to converge to the optimal λ

Sequence	Rate (kb/s)	Find the initial λ	Find the optimal λ
Akiyo (CIF)	576	2.96	2.43
	1536	3.00	2.42
Coast Guard (QCIF)	384	2.87	3.17
	896	2.96	3.90

- For the frames after the first frame in the video sequence or in a new scene: The contents of the frames in the same scene share great similarity, and we can make use of this feature to set the proper initial λ value and reduce the time of iteration. Table 4 below shows some statistical examples of how similar the optimal λ values are for the consecutive frames under different rates for different sequences. For the Akiyo sequence under 1.64 Mb/s, 63% of the I-frames and 69% of the P-frames will get the same optimal λ value as its preceding I-frame and P-frame, respectively. For both the I and P frames, their optimal λ values will not exceed the 15% range from the optimal λ value of their preceding frames. This indicates the initial λ value of the current frame can be set as the optimal λ value of the previous frame with the same coding type.

Table 4. Difference of the optimal λ values for the consecutive frames with the same coding type.

		I Frames				P Frames			
Sequence	Rate (kb/s)	Same	<10%	<15%	Other	Same	<10%	<15%	Other
Akiyo (CIF)	576	63%	0	37%	0	69%	26%	5%	0
	1536	58%	42%	0	0	74%	22%	0	4%
Coast Guard (QCIF)	384	N/A (since only one I frame in the sequence)				62%	22%	13%	3%
	896					8%	69%	18%	5%

- Bypass the iterations when transport the pre-stored video

  For the transport of pre-stored FGS video streams, all the enhancement layer bit streams are stored on the server. They will be truncated at the delivery time according to the available bandwidth. When generating the enhancement layer bit-streams of each frame offline, we can also produce a group (rate, λ) tuples, indicating the optimal λ values under different rate points. At the transport time, all we need to do is to find a nearest rate point to the bandwidth requirement, and use the corresponded λ value as the optimal one.

3.3. Simulation Result

We perform simulations to show the effectiveness of the proposed enhancement layer truncation scheme. The sequences of "Akiyo" and "Bicycle," both in the CIF

format, as well as the sequences of "News" and "Coast Guard," both in the QCIF format, are used in the simulation to compare the performance of "even truncation" with our algorithms. The base-layer is encoded with the quantization parameter of $Q = 31$ for both I frames and P frames. There is no B frame in the sequence. The threshold to stop the iteration is *TH=5%* of the frame bit-budge, and the maximum iteration number is set to 10.

To evaluate the coded video quality of the proposed algorithms, we adopt two quality measures: PSNR (Peak Signal to Noise Ratio) and Intra-Frame Quality Variation. *PSNR* is defined as:

$$PSNR_I = 10\log_{10}\left(\frac{255^2}{MSE_I}\right) \tag{9}$$

where

$$MSE = \frac{1}{M \times N}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|x(m,n) - \hat{x}(m,n)\right|^2 \tag{10}$$

is the *Mean Squared Error* between the decoded picture and its original representation in the video sequence. In Equation 10, $x(m,n)$ and $\hat{x}(m,n)$ are the original and reconstructed pixel values at the spatial location (m,n) of the picture, respectively. The picture has M lines, and each line has N pixels.

We use Equation 11 to measure the intra-frame quality variation, where K is the total number of luminance macro-blocks in the frame, $\overline{MSE}$ is the average value of the mean square error of all the luminance macro-blocks. It is actually the variance of each luminance macro-block's mean square error.

$$IQV = \frac{1}{K}\sum_{i=1}^{K}(MSE_i - \overline{MSE})^2 \tag{11}$$

Table 5 below shows which enhancement layers will be truncated for the above video sequences under different bit-rates. It also points out how much the truncated enhancement layers will remain compared to the size of the original enhancement layer.

Table 5. Truncated layers and remained portion of the truncated layer.

Sequence	Bit rate (kbps)	Truncated Enhancement Layer	Remained Portion of the Truncated EL (%)
Akiyo	576	3	*52*
	1536	4	*58*
Bicycle	640	2	*46*
	1920	3	*32*
Coast Guard	384	3	*43*
	896	4	*38*
News	384	3	*50*
	896	4	*57*

Table 6 shows the average PSNR of the whole sequence obtained from different truncation schemes. Our simple re-encoding scheme loses the coding gain in terms of PSNR up to about 0.14 dB in the Akiyo sequence at 576 kb/s. For other cases, the simple re-encoding method could lose the coding gain up to about 0.3 dB. This is because it drops "1" bits in every block from the highest AC coefficients, and does not consider their corresponding distortions. The rate-distortion optimization scheme improves the PSNR performance from about 0.14 to 0.56 dB. Another conclusion is that, the more the truncated enhancement layer remains, the more gain we can achieve by the R-D optimization.

Table 6. Performance of the proposed schemes and even truncation: PSNR (dB).

Sequence	Bit rate (kbps)	Even Truncation	Drop 1	*Gain*	R-D Optimization	*Gain*
Akiyo	576	35.36	35.50	*0.14*	35.85	*0.49*
	1536	39.64	39.59	*-0.05*	40.07	*0.43*
Bicycle	640	25.96	25.86	*-0.10*	26.18	*0.22*
	1920	29.05	28.90	*-0.15*	29.20	*0.14*
Coast Guard	384	30.33	30.21	*-0.12*	30.62	*0.29*
	896	34.38	34.10	*-0.28*	34.56	*0.18*
News	384	31.47	31.38	*-0.09*	31.85	*0.38*
	896	36.36	36.06	*-0.30*	*36.92*	*0.56*

Table 7 shows the average Intra-frame Quality Variation (IQV) of the whole sequence obtained from different truncation schemes. The reduction of the IQV between the proposed schemes and "even truncation" is also shown in the table. We can see that our simple re-encoding scheme, although losing some coding gain in terms of PSNR, can drop the IQV from 12% to 53%, while the rate-distortion optimization scheme can further improve the performance from 37% to 80%.

Table 7. Performance of the proposed schemes and even truncation, in terms of IQV.

Sequence	Bit rate (kbps)	Even Truncation	Drop 1	*Reduction (%)*	R-D Optimization	*Reduction (%)*
Akiyo	576	545	398	*26.8*	339	*37.8*
	1536	60.9	53.5	*12.1*	30.7	*49.6*
Bicycle	640	12995	8019	*37.9*	5974	*54.2*
	1920	2357	1091	*53.6*	877	*62.9*
Coast Guard	384	1434	1013	*22.9*	550	*62.0*
	896	160	77.7	*33.8*	50.5	*79.2*
News	384	1946	1463	*31.4*	877	*58.8*
	896	172.4	114.2	*33.6*	*46.2*	*73.2*

Figure 14 shows the PSNR improvement for each frame in the Akiyo sequence at 576 kb/s. For the whole sequence, our algorithm of reallocating the number of bits to each block and simply dropping the "1" bits to meet the new bit-target, as described in Section 3.2, can obtain an average of 0.19 dB PSNR improvement. After the R-D optimization, an average PSNR improvement of 0.49 dB can be achieved.

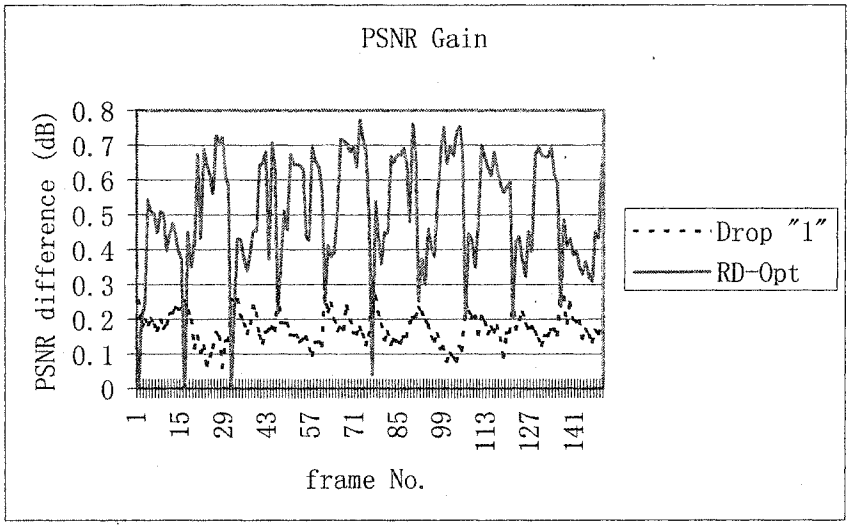


Figure 14. PSNR of each frame (up) and PSNR improvement (down) in the Akiyo Sequence (at 576 kb/s).

Figure 15 illustrates the intra-frame quality variation for each frame (up) and the reduction of the quality variance (down). Since our method can improve the quality of each block uniformly, for the Akiyo sequence, it reduces the intra-frame quality variation by 27% if we simply drop the "1" bits to meet the new bit-target, and 37% after the R-D optimization.

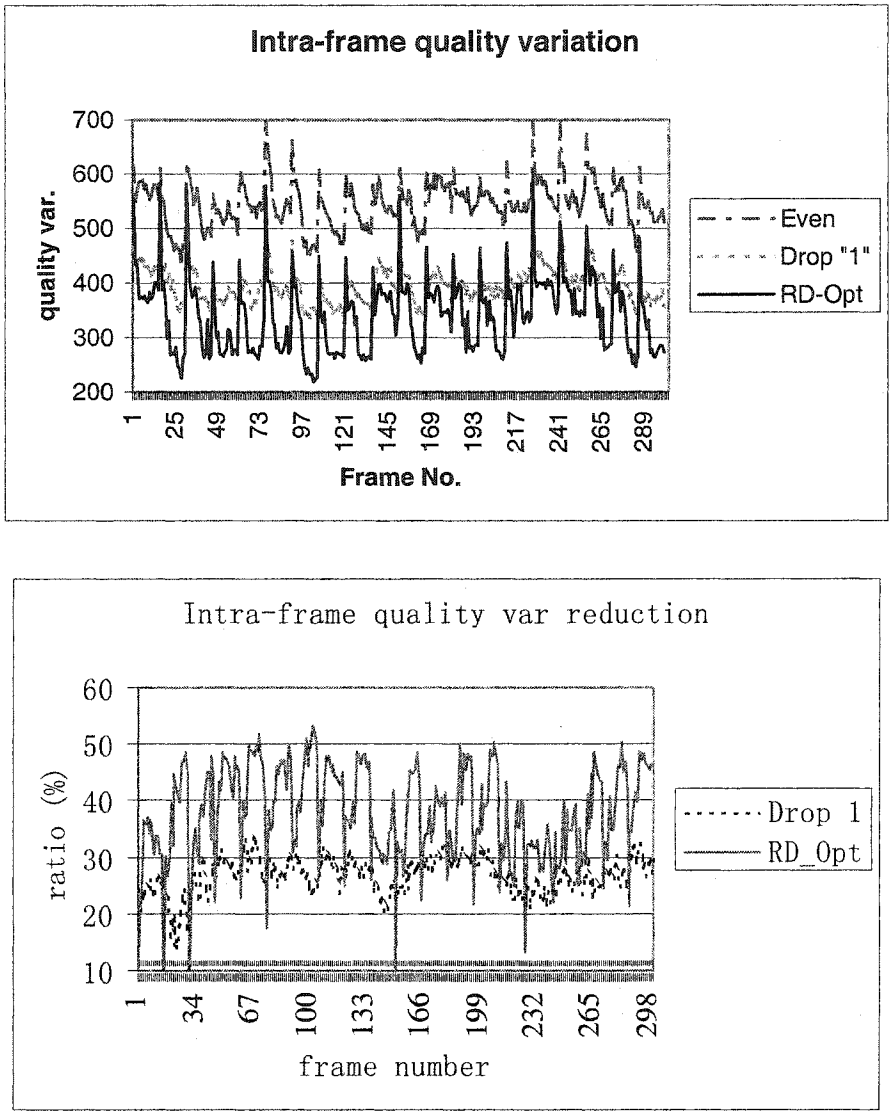


Figure 15. Intra-frame quality variance (up) and variance reduction (down) in the Akiyo Sequence (at 576 kb/s).

Figure 16 shows the decoded frame 61 by the "even" and "R-D optimized" truncation algorithms. It can be seen that our algorithm also achieves better subjective visual quality.

Figure 16. Subjective visual quality of the decoded frame 61 from Even Truncation (left) and R-D optimization (right) (at 576 kb/s).

3.4. Summary

In this chapter, we studied the rate adaptation problem for the FGS enhancement-layers. First, we point out that the original "even truncation" scheme will lead to both the inter-frame quality variation and the intra-frame quality variation. Second, we propose a standard-compatible method that can redistribute the available bit-budget for the last transmitted bit-plane to each block. Third, we solve the truncation problem as an optimized bit-dropping problem using the Lagrange Multiplier algorithm. We also suggest to use the variance of the Mean-Square-Error values for each macroblock in a frame as a criterion to measure the intra-frame quality variation. Our proposed schemes enhance the whole frame more uniformly and the intra-frame quality variation can be

reduced both objectively and subjectively. Simulation results show the effectiveness of the proposed algorithms.

Chapter 4. Multi-path Transport of the FGS Video

4.1. Introduction

For today's Internet video streaming applications, one important concern is to dynamically deliver video contents to users with different available resources, access networks, and interests. Video contents need to dynamically adapt to the conditions of different users. FGS [16] has been proposed in MPEG-4 to meet this requirement. For the two bit streams generated by an MPEG-4 FGS encoder, the base-layer stream provides the basic visual quality, and the enhancement stream is used to improve the base-layer quality. The purpose and importance of the two streams are different. The enhancement information is useless without the correct decoding of the base-layer information, thus the base-layer should be strongly protected. The enhancement stream is composed of several enhancement layers with a bit-plane coding scheme distinguishing FGS from the traditional scalability. FGS is capable of providing continuous rate-control for the enhancement stream since the enhancement layers can be truncated at any point to achieve the target bit-rate.

As mentioned before, most research efforts of FGS have focused on how to improve its coding efficiency [40][41], how to truncate the enhancement layers to minimize the quality variation both between the adjacent frames [42][43][44][54][55] and within a frame [45][46][47], and how to modify the FGS coding structure to add temporal [56] and spatial [57] scalability. How to protect the FGS stream has also been well addressed. The error control and error concealment techniques introduced in [49] and

error resilience coding techniques described in [50] can be used for the base-layer coding and transmission. For the enhancement layer bit-stream, Wang [51] and Van der Schaar [52] propose to use the adaptive FEC or unequal error protection to protect different enhancement-layers according to their importance when transporting the FGS bit streams over lossy channels.

A big challenge for video transmission is the instantaneous congestion problem from the network. Theoretically, FGS can dynamically adapt to available network bandwidth estimated using some end-to-end network probing. However, the network probing usually cannot reflect instantaneous congestion effectively because the probing response time is often longer than instantaneous congestion duration. FEC-based approaches and re-transmission-based approaches can somehow recover packet-loss from the instantaneous congestion but they may possibly aggravate the congestion by introducing more traffic.

Multi-path approach is a promising solution to the instantaneous congestion problem. Multiple paths may be available in the network for many streaming video applications [63][64][65]. In the Internet, there may be multiple paths available between the sender and the receiver as illustrated in Figure 17. Particularly in recent years, numerous overlay network technologies were proposed such as CDN (Content Distribution Network), Peer-to-Peer network, and Ad-hoc network [64][65]. As an example, in peer-to-peer streaming, multiple paths can be provided for transmitting the content to the client side. In wireless networks, MIMO (Multiple Input Multiple Output) systems can be used to transmit the video content over multiple channels [66], where the

video bit-stream is first interleaved in the group-of-block (GOB) level and then transmitted in the MIMO channels. In these cases, each path may have lower bandwidth, but the total available bandwidth is higher than the single-path case. Multi-path transport can also improve the transport reliability by overcoming the instantaneous congestion problem often encountered in the single-path case.

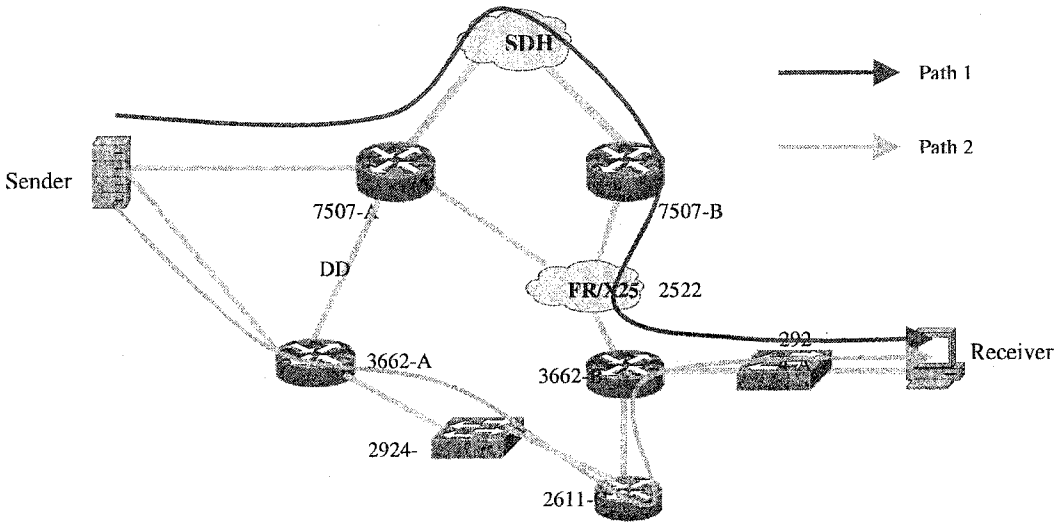


Figure 17. Illustration of the multi-path video streaming

One thing we should mention here is that, we assume the multi-path established will have as little overlapping as possible, so that the congestions in one path will have minimum impact to the other path. Another assumption we take here is that, the instantaneous congestion occurs in the routers except the one directly connected to the end-user, otherwise, there is no way we can circumvent the congestion by introducing an extra path.

Multiple Description Coding (MDC) has been proposed for transporting a video stream over multiple independent channels. In MDC, different descriptions of the

original video content are transmitted via different parallel channels. Since the error events of different channels are independent, the probability that all channels simultaneously experience losses is low. A comprehensive review of MDC is given in [69], and some MDC schemes on single-layered video coding have been introduced in [50]. Research on using MDC for transporting video streams over multiple wireless channels has been discussed [67][68]. Multiple description coding addresses the problem of unreliable channels by means of sending independent descriptions over multiple channels, while for layered coding, like FGS, it address the problem of heterogeneous end-user or network condition by means of bit-stream truncations. Some research have been conducted in order to achieve both the transport robustness to the unreliable channel and end-user requirement. For the transport of the pre-stored single layered video bit-stream, Reibman [71] proposes to split the DCT coefficients in a block into two channels. Any DCT coefficients larger than a threshold are duplicated into each available channel, while the rest coefficients will be alternatively separated into each channel. In [27], Chou introduced a scheme combining layered coding with MDC to transmit MPEG-4 FGS streams over multiple channels. In Chou's scheme, video sequence is divided into group of frames (GOF). Different layers in a frame are protected by different Reed-Solomon (RS) codes according to the importance of the layer, and multiple descriptions are formed by taking one byte from every RS coded layer of every frame in a GOF. The distortion on the receiver side can be minimized by carefully allocating different rates to different layers under given channel conditions. However, this scheme is designed for the

multicast applications by broadcasting different descriptions into different channels simultaneously, it requires as much as 32 channels to carry all the descriptions.

In this chapter, we propose a point-to-point transport scheme to deliver MPEG-4 FGS streams over multiple channels. In our scheme, the base-layer and lower enhancement layers (which have lower bit-rates) are protected and duplicated in the multiple paths, and higher bit-rate enhancement layers in the original FGS stream are split into multiple descriptions for effective transport over the multiple paths in the network. Several techniques are proposed to improve the video quality. Simulation results show the effectiveness of the proposed approaches.

The rest of this chapter is organized as follows. Section 4.2 first describes the statistical properties of FGS streams and our FGS transport scheme over multiple channels, then it presents our enhancement-layer splitting mechanism. Simulation results are shown in Section 4.3 and the conclusion is drawn in Section 4.4.

4.2. FGS Transport with Multi Paths

4.2.1. Statistics of FGS Streams

Before designing the streaming mechanism for the FGS video streams, we should first pay attention to some statistical characteristics of the encoded FGS bit-streams. We encoded the "Coast Guard" and the "Akiyo" sequences (both in the CIF format) by setting the quantization parameter of the base-layer to Q=31 for both I frames and P frames, there is no B frame in the sequence.

Figure 18 shows the result for the "Coast Guard" sequence. For the base-layer and some of the enhancement-layers (EL), such as EL1 and EL2 (which carry the information of large residual errors), the bit-rate is relatively low (only about 243 kb/s and 54 kb/s). On the other hand, from EL3, the bit-rate begins to grow rapidly, and the average PSNR of the entire sequence continues to increase significantly. For instance, the EL3 bit-rate is about six times of the base-layer bit-rate, and the EL7 bit-rate is over eighteen times of the base-layer bit-rate.

Figure 19 shows the result for the "Akiyo" sequence, which has less movement and simple texture. Similar conclusions can be drawn. We also studied many other standard video testing sequences and obtained similar statistical results as those for the "Coast Guard" and the "Akiyo" sequences.

The above statistical results give us some insight on how to transport FGS video streams. Because of the small bandwidth requirement, it is reasonable to use forward error correction (FEC) or other error protection schemes [49][50] to protect the base-layer and some lower enhancement layers (Such as EL1) of FGS videos. However, using the schemes proposed in [51] or [52] to protect the high bit-rate enhancement-layers may not be effective. Due to the high bit-rate and that more overhead-bits are required to protect the high bit-rate enhancement-layers, they may be dropped when the channel bandwidth is limited. The highly fluctuated network bandwidth may lead to low decoded video quality.

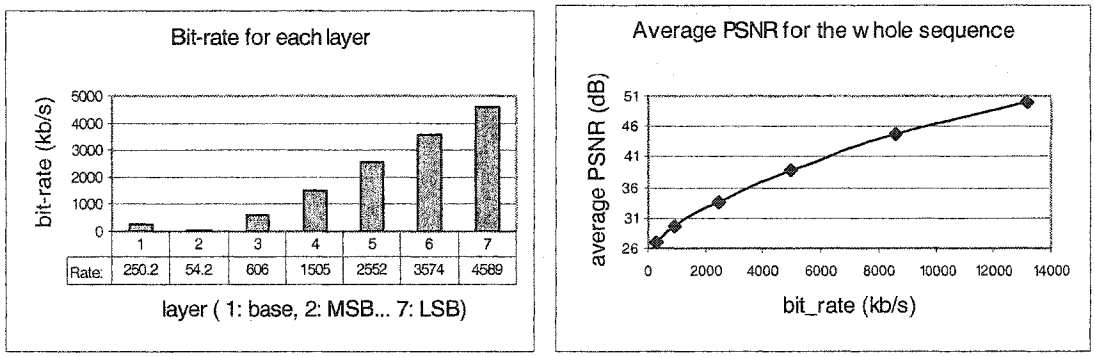


Figure 18. Bit-rate for each layer in the "Coast Guard" Sequence (left) and the corresponding average PSNR of the entire sequence

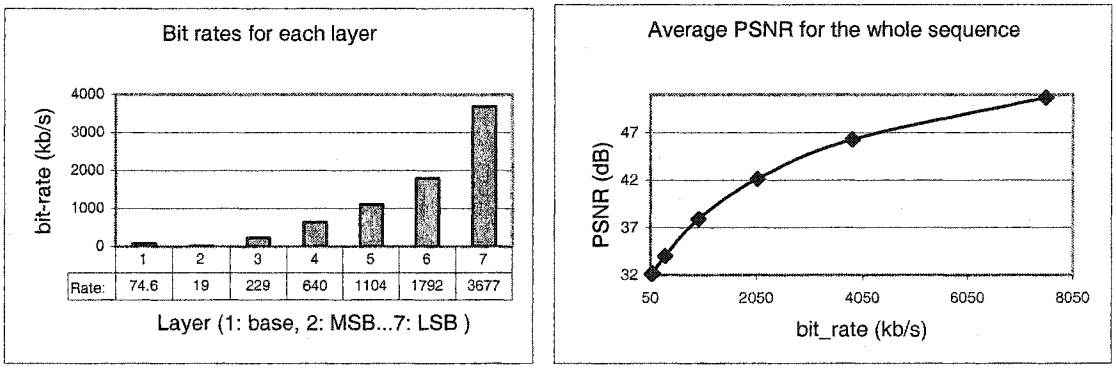


Figure 19. Bit-rate for each layer in the "Akiyo" Sequence (left) and the corresponding average PSNR of the entire sequence

For the network side, video transport should be "TCP-friendly" to obey the fairness policy of the Internet. Because applications share network resources, the network often encounters instantaneous congestions from time to time. This will cause burst packet-losses. The congestion problem also means that the more bandwidth required from the applications, the more difficult for the network to support the Quality of Service (QoS) for such kind of applications. The bit-rates of some higher enhancement layers of FGS video seem to be too large for today's Internet. However, if we can divide a large

FGS enhancement layer into several sub-streams, it may be easier for the network to find several lower bandwidth paths than to find a single large bandwidth path. In addition, a multi-path solution can also improve the video transport resilience to burst packet-losses. This is the basic intuition behind our proposed scheme.

4.2.2. Multi-path Transport for an FGS Stream

With the help of multi-paths between the server and the client, the channel capacity or the diversity can be improved. For the Internet, there are two extreme cases: one is to duplicate the original video content into each available channel. This will help to improve the robustness of the video transport, since a lost packet in one channel can be recovered from other channels. However, if the bandwidth of each channel is not sufficient, the duplicated bit-streams will result in low-quality video. The other extreme case is to segment the original video stream into non-overlapped packets, and cast them to different channels. This method can improve the effective transport bit-rate, but if packet-loss occurs in any channel, the corresponding information will not be able to be reconstructed. Thus, taken into consideration of both the importance and the bit-rate of each layer, we propose the following FGS multi-path transport scheme:

- Base-layer and certain enhancement layers with lower bit-rates are protected using traditional forward error correction schemes. This part of the stream is copied to every available path and delivered to the client side. In this way, the base-layer is strongly protected, and the enhancement layers with low bit-rates carrying larger residual errors are also well protected.

- Enhancement layers with higher bit-rates are converted into multiple descriptions to take advantage of the multi-path transmission approaches. Each path carries one split description of the original higher enhancement layer. In this way, each split enhancement-layer has a lower bit-rate compared with its original version, thus making it possible to be transmitted in one path and easier to be protected. What's more, when one video packet is lost, similar information is possible to be retrieved from other paths to improve the decoded video quality.

4.2.3. Splitting Mechanism for the FGS Enhancement Layer

Spliting the high bit-rate enhancement layers is a procedure to redistribute the "1" bits in the original enhancement-layer blocks into different descriptions within the available channel bandwidth. A simple method is to distribute the "1" bits evenly into the multiple descriptions, then each description is entropy encoded with the original VLC table. For example, if two paths are available, and in the original block, the 64 bits are: 000011000101...110, the "1" bits will be evenly distributed into two descriptions as:

Original block:	000011000101...110
Description 1:	000010000100...100
Description 2:	000001000001...010

Another possible scheme is to distribute the "1" bits according to their positions in the original block and their corresponding importance to the human visual system. For example, the coefficients belonging to the upper-left corner in a block represent more

important low frequency components, thus the corresponding "1" bits can be copied to all the descriptions, and the rest "1" bits can be assigned uniformly to each description.

The diagram to process the enhancement layer is shown in Figure 20 below.

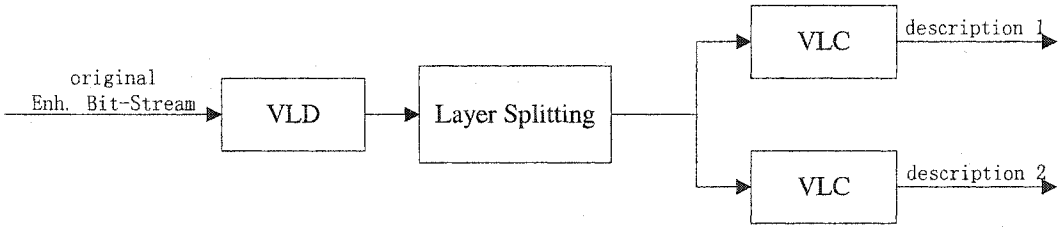


Figure 20. Diagram of Splitting the FGS Enhancement Layer

4.2.4. Improved Splitting Mechanism

The splitting scheme described above is a simple one to generate balanced descriptions with similar bit-rates and decoded quality. This can be seen from Table 8 below. After the splitting, the rate required to encode the whole enhancement layer in a frame will be reduced. However, if the channel bandwidth is not enough to support the split layer, it will also need to be truncated, thus leading to the quality inconsistency within the frame area.

Table 8. Performance of simple enhancement layer splitting

Sequence	Layer to be split	Split Bit-rate (kb/s)		Decoded PSNR (dB)		Original Layer Bit-rate (kb/s)
		Des 1	Des 2	Des1	Des 2	
Akiyo	EL 3	448	466	36.1	36.2	640
	EL 4	824	824	39.7	39.6	1104
Bicycle	EL 3	1560	1544	29.8	29.7	2200
	EL 4	2520	2496	34.6	34.5	3280
Coastguard	EL 3	304	302	31.9	31.8	430
	EL 4	537	531	34.9	34.7	709
News	EL 3	270	269	31.9	31.8	373
	EL 4	444	437	36.4	36.2	570

Rate-distortion optimization helps to solve the mentioned problem. Generally speaking, if two paths are available, a "1" bit in an enhancement layer can be assigned to the 1st description, or to the 2nd description, or to both of them, or to neither of them. This results in different rates and distortions for each description. Therefore, similar to what is done in Chapter 3, we can use the trellis shown in Figure 22 to search each block for the optimal bit redistribution. For each block, when a "1" bit is encountered in the block, it introduces a new stage. "10," "01," "11," and "00" are four states used to indicate the "1" bit is assigned to the 1st description, to the 2nd description, to all the descriptions, and to none of them. For each state, there will be 4 incoming routes from the previous stage. Each route will have a cost function, similar to that described in [70], $J(\lambda) = D(R_1, R_2, P_{loss}) + \lambda(R_1 + R_2)$, where R_1 and R_2 are the numbers of bits produced up to the current stage in each description, P_{loss} is the packet-loss probability parameter and is related to the network status, $D(.)$ is the overall distortion of the two descriptions

associated with the rate and the network status. By comparing the 4 cost functions, one optimal way will be kept. This process goes on until the bit-budget for each description is met.

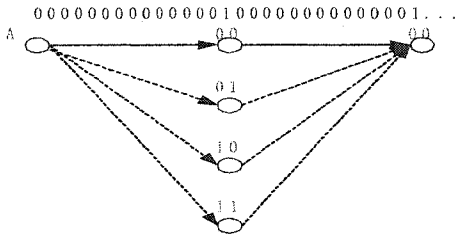


Figure 21. Trellis search of FGS enhancement layer splitting with R-D optimization

The full optimization process described above can improve the video quality. However, it introduces a huge extra computational burden. The total searching routes in one block is analyzed as $3n^2+1$, where n is the number of "1" bits in the block. In order to reduce the extra computation, we can apply the rate-distortion optimization on the split layers individually. This process can be illustrated in Figure 22 below.

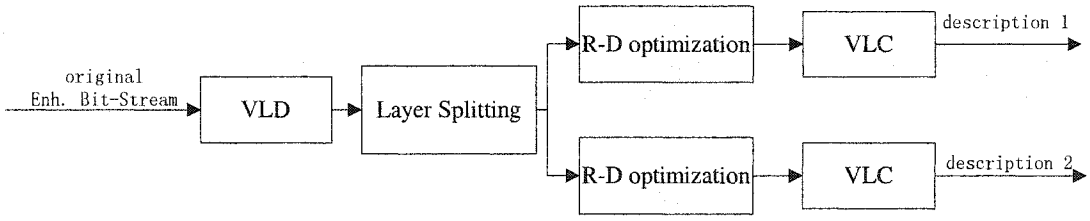


Figure 22. Diagram of Splitting the FGS Enhancement Layer with individual R-D optimization

The extra computation incurred by individual R-D optimization is a linear function of the number of "1" bits in each block. However, after the enhancement layer block is split, the "1" bits tend to be sparsely distributed. Symbols with long run will be

generated, and the efficiency of the R-D optimization process will be affected. In order to compensate for the symbols with long run, we design the following procedure: after the R-D optimization of one description, the "1" bits in the other description can be added back to split a codeword with a long run into two codewords with shorter runs, if the bits to encode the two new codeword is no more than that for the original codeword. In this way, we not only reduce (or keep) the required rate in one description, but also reduce the distortions in both descriptions. One example of adding "1" bit from another description is shown in Figure 23. After this step, the rate for one description may be reduced, and we can make use of the saved bit-budget to add more information, i.e., the "1" bits dropped during the optimization, back to the current description. The whole process to split the enhancement layer is illustrated in Figure 24.

Des. 1 after R-D optimization	0	0	1	0	0	0	0	0	0	1
					⇑	Add "1" from description 2				
Des. 2: before R-D optimization	0	0	0	0	1	0	0	1	0	0

Figure 23. Example of Splitting the FGS Enhancement Layer with cross-description R-D optimization

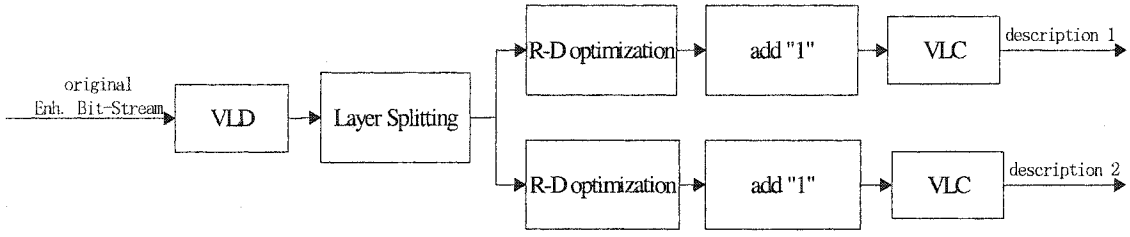


Figure 24. Diagram of Splitting the FGS Enhancement Layer with cross-description R-D optimization

4.3. Simulations Results

In this section, we first examine the performance of the enhancement-layer splitting schemes under channels without packet loss, since that is the best quality we can receive. We then evaluate their behaviors under a packet loss environment.

4.3.1. Splitting Schemes under idea channels

The "News" sequence (QCIF format) is used in the following experiments. The base-layer is quantized with the quantization parameter $Q = 31$ for both I frames and P frames, no B frame appears in the stream.

In experiment 1, three enhancement layers are generated, where the first two enhancement layers, as well as the base-layer, are copied into two individual paths, while the third enhancement layer is split into two descriptions.

As an example, we assume that the channel rate can only support up to ¾ split enhancement-layer 3 (EL3), with the corresponding total rate of 397 kb/s. We compare the PSNR and the intra-frame quality variation in two scenarios:

1. Sequence decoded by the base-layer, EL1, EL2, and the *truncated* **SPLIT** EL3 up to the required rate;
2. Sequence decoded by the base-layer, EL1, EL2, and the **R-D optimized** *split* EL3 (shown in Figure 25) at the required rate.

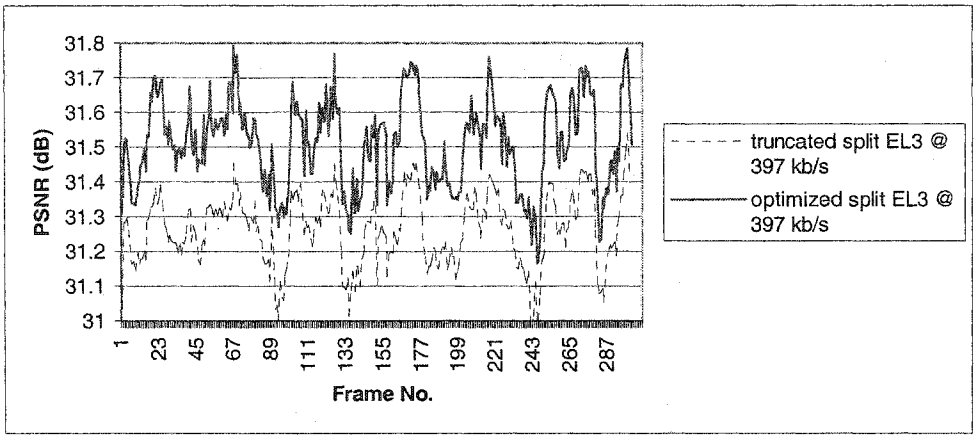


Figure 25. PSNR of the decoded frames at 397 kb/s

Figure 25 above shows the PSNR of the decoded frames in different scenarios. Combining the proposed approaches, the PSNR for the whole sequence can be improved by about 0.26 dB. Figure 26 shows the PSNR difference.

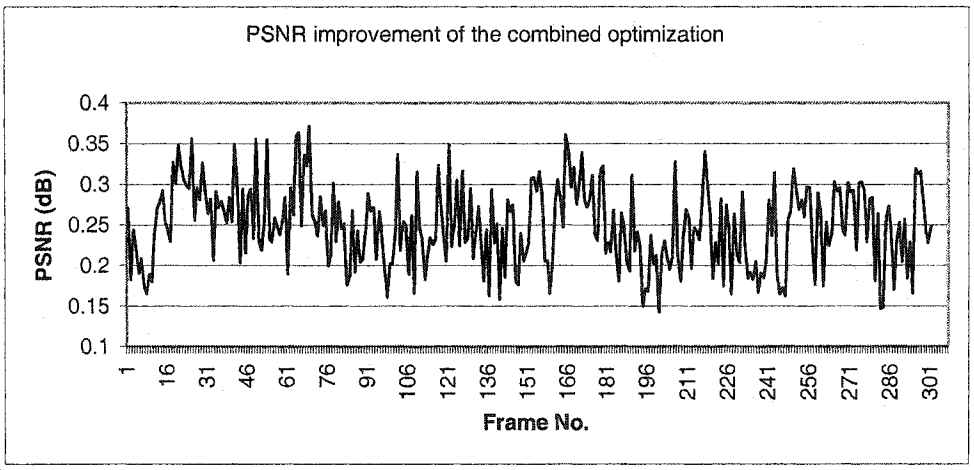


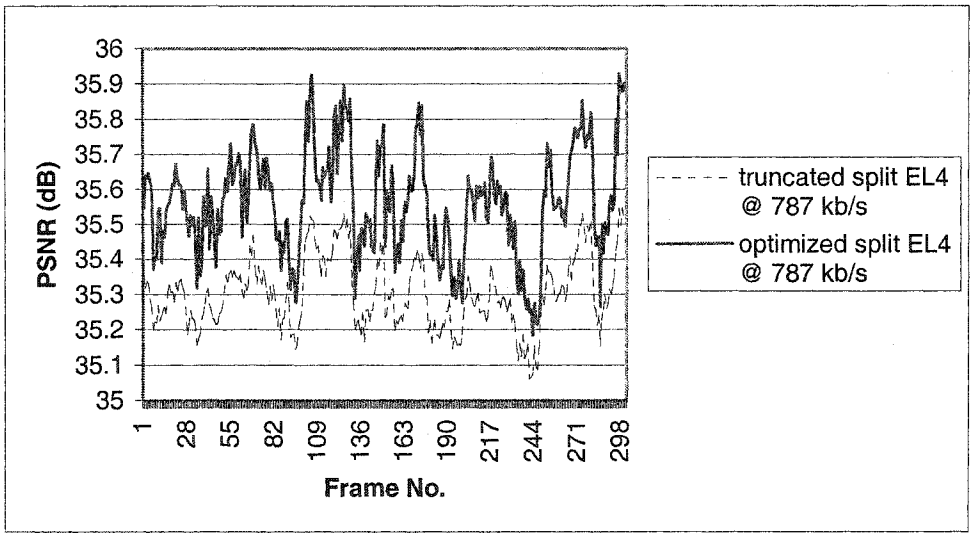
Figure 26. PSNR difference between the optimized split EL3 and truncated split EL3 at 397 kb/s

A similar conclusion can be drawn from experiment 2, where four enhancement layers are generated. The first three enhancement layers, as well as the base-layer, are

copied into two individual paths, while the fourth enhancement layer is split into two descriptions.

Assuming that the channel rate can only support up to ½ and ¾ split EL4, with the corresponding total rate of 787 kb/s and 896 kb/s, we compare the PSNR and the intra-frame quality of the following 3 sequences:

1. Sequence decoded by using the base-layer, EL1, EL2, EL3, and the *truncated* **SPLIT** EL4 up to the required rate;
2. Sequence decoded by using the base-layer, EL1, EL2, EL3, and the **R-D optimized** *split* EL4 (shown in Figure 25) at the required rate..



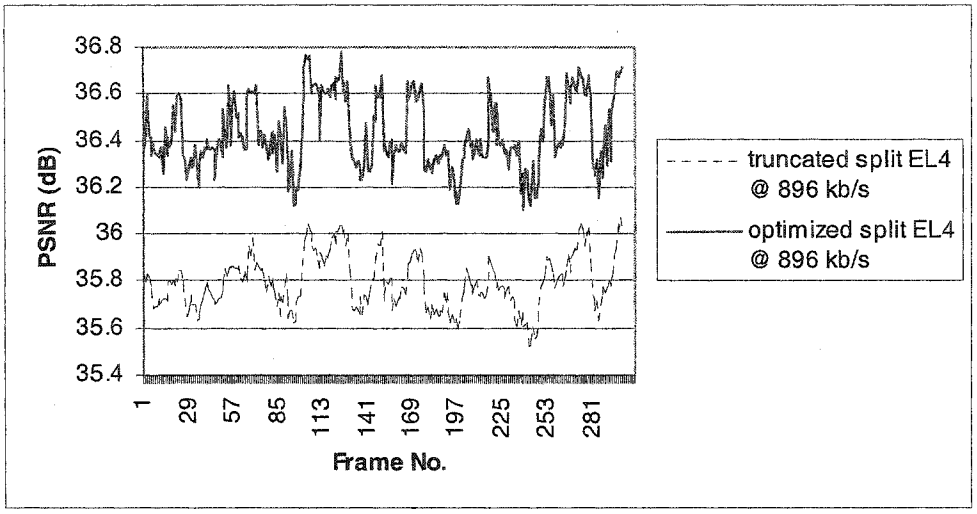


Figure 27. PSNR of the decoded frames at 787 kb/s (up) and 896 kb/s (down)

Figure 27 above shows the PSNRs of the decoded frames in different scenarios under different rates. With our combined approach, the average PSNR for the whole sequence can be improved by up to 0.26 dB at 787 kb/s, and 0.64 dB at 896 kb/s. Figure 28 and Figure 29 show the PSNR difference for the different cases.

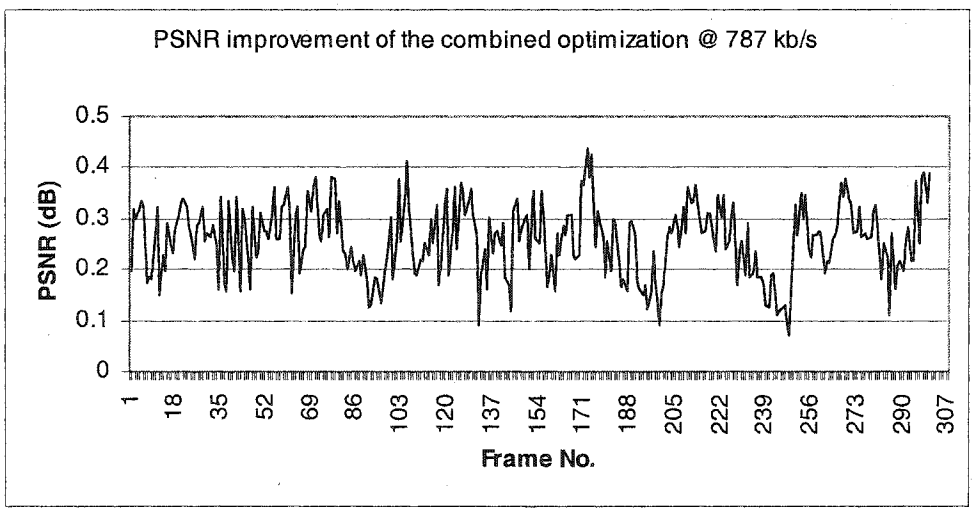


Figure 28. PSNR difference between the optimized split EL4 and truncated split EL4 at 787kb/s

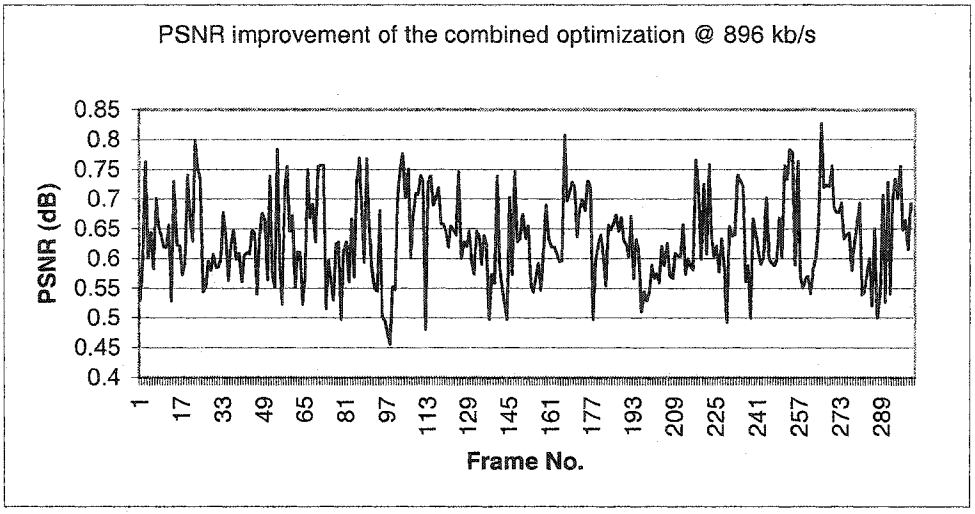


Figure 29. PSNR difference between the optimized split EL4 and truncated split EL4 at 896 kb/s.

The data above is summarized in Table 9. We also list the intra-frame-quality variation (IQV) for different sequences in Table 10, in which we can see the IQV is also reduced from 30% to 50% after the combined optimization under different bit-rates.

Table 9. PSNR Performance of enhancement layer splitting for description 1

Split Layer	Rate (kb/s)	Truncated split layer (dB)	Combined approach (dB)	Gain (dB)
EL 3	397	31.26	31.52	0.26
EL 4	787	35.30	35.56	0.26
EL 4	896	35.79	36.43	0.64

Table 10. Intra-frame quality variation (IQV) performance of enhancement layer splitting for description 1

Split Layer	Rate (kb/s)	Truncated split layer	Combined Approach	Reduction (%)
EL 3	397	1508	1013	32.8
EL 4	787	152.8	107.9	29.4
EL 4	896	115.9	59.4	48.8

4.3.2. Performance in packet-loss environment

We use the same coding condition for the "News" sequence in this section. We assume that the base-layer can be delivered to the client side without any damage. For a fair comparison between the single channel approach and a multi-path approach, we assume the channel can hold the entire enhancement layers. Four enhancement layers are generated, where the base-layer and the first two enhancement layers are copied into two individual channels to be delivered to the client side, and the last two enhancement layers are split into two descriptions. No FEC or ARQ is added to the bit streams. The encoded bit stream is packetized with the packet size of 500 bytes. Table 11 below shows the bit rate for each layer that the channel should hold.

Table 11. Bit-rate for each layer in each channel (kb/s)

		Base	EL1	EL2	EL3	EL4	Total
Multi-path	Channel1	22.8	18.5	178	270	444	933.3
	Channel2				269	437	925.3
Single-path					373	570	1162.3

We use a two-state Markov channel model as shown in Figure 30 to simulate the packet-loss states of the Internet channel. The "Good" state is the state in which channel packet-loss probability is low, while the "Bad " state indicates that the channel is in the burst error period. p_1 is the transition probability of the channel to change from the "Good" state to the "Bad" state, while p_2 is the probability that the channel remains in the "Bad" state. In the "Good" state, the packet drop rate is P_{on}, and in the "Bad" state, the packet drop rate is P_{off}. We also use two sets of parameters to simulate different channel conditions, as depicted in Table 12.

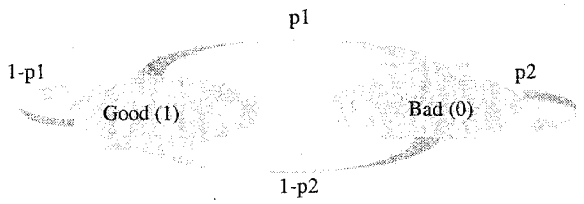


Figure 30. Two-state Markov channel model

With the two-state Markov model, we performed 3 groups of experiments with different combinations of the two channel conditions. The conditions are described in Table 13.

Table 12. Experiment models for multi-path transmission

	p_1	p_2	p_{on}	p_{off}
Mod1	0.2	0.7	0.01	0.25
Mod2	0.4	0.8	0.01	0.4

Table 13. Experiment conditions for multi-path transmission of the FGS bit-streams

	Multi-Path	Single-Path
Experiment 1	Mod1+Mod2	Mod1
Experiment 2	Mod1+Mod1	Mod1
Experiment 3	Mod2+Mod2	Mod2

In experiment 1, we assume that in the multi-paths, one channel is in a relatively good condition, while the other channel is in a relatively congested condition. Each channel needs to hold the duplicated base-layer, EL1, EL2 and the split EL3, EL4, which has a total bandwidth of 933 kb/s. As a comparison, the single path is in a good condition, and it will hold the base-layer, EL1, EL2 and EL3, which has a total bandwidth of 1162 kb/s. Figure 31 shows that the multi-path approach can achieve an average of 2.0 dB gain in PSNR for the entire sequence.

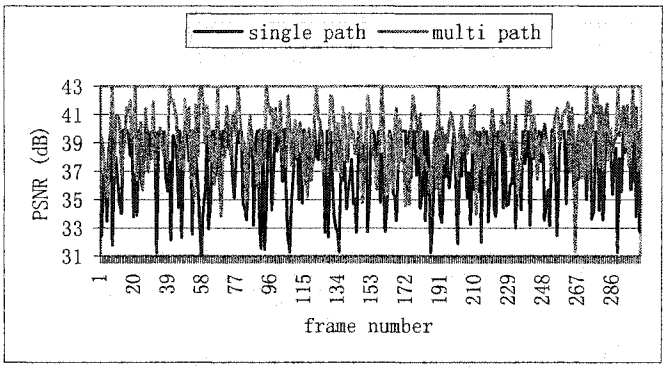


Figure 31. Performance analysis between the multi-path transmission approach and the single path transmission approach, experiment 1.

In experiment 2, we assume that in the multi-paths, both channels are in a relatively good condition, with a bandwidth of 933 kb/s each, and the single channel is

also in a good condition, with a bandwidth of 1162 kb/s. Figure 32 depicts that the multi-path approach can achieve an average of 2.4 dB gain in PSNR for the entire sequence.

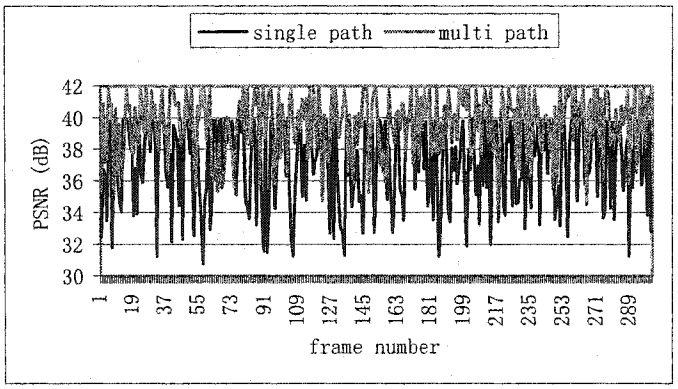


Figure 32. Performance analysis between the multi-path transmission approach and the single path transmission approach, experiment 2.

In experiment 3, we assume that in the multi-paths, both channels are in a relatively congested condition, and the single channel is also in a congested condition. **Error! Reference source not found.** illustrates that the multi-path approach can achieve an average of 2.6 dB gain in PSNR for the entire sequence.

When the channel in the multi-path approach cannot hold the whole split layer, truncation is need. For this case, we did the following experiment, where three enhancement layers are generated. In the transmission, the base-layer and the first enhancement layer are copied into two individual paths, while the last enhancement layer is split into two descriptions at the bit-rate of 397 kb/s. We also perform three groups of experiments with the conditions shown in

Table 13. The average PSNR of the whole decoded sequence is shown in

Table 14. There is still a gain of about 0.5-0.7 dB if we deploy the combined optimization when we split the enhancement layers.

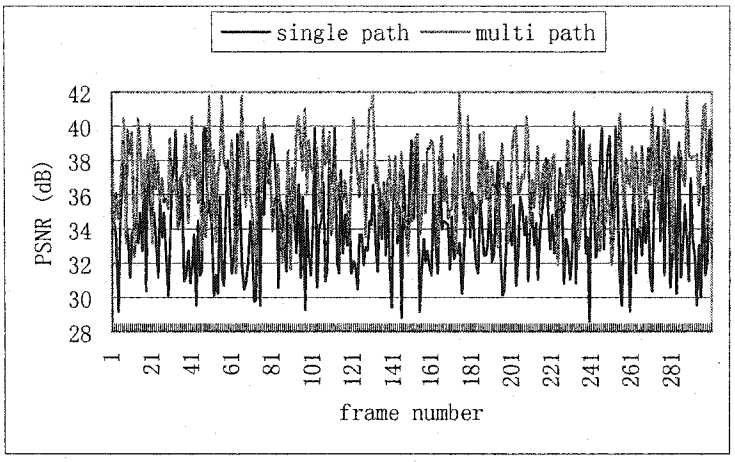


Figure 33. Performance analysis between the multi-path transmission approach and the single path transmission approach, experiment 3

Table 14. Average PSNR of the whole sequence (dB)

Experiment Condition	Truncate the split EL	Combined optimized split EL
Experiment 1	29.7	30.4
Experiment 2	28.8	29.5
Experiment 3	27.8	28.3

4.4. Summary

In this chapter, we discuss the problem of transmitting a FGS stream over multiple paths. In order to overcome the problems of instantaneous network congestion

and large bandwidth requirement from higher FGS enhancement layers, we propose a new approach that utilizes the multi-path features to transmit the split version of the high FGS enhancement layers. We also propose solutions to split the enhancement layer into multiple descriptions, first is the straightforward splitting scheme, then follows the rate-distortion optimization schemes. Simulation results show that splitting the enhancement layers, together with the multi-path transport approach, can improve the end-to-end FGS video streaming quality.

Chapter 5. MPEG Video Streaming with VCR Functionality

5.1. Introduction

A video streaming system should be capable of delivering concurrent video streams to a large number of users. The realization of such a system presents several challenges, such as the high storage-capacity and throughput in the video server and the high bandwidth in the network to deliver large number of video streams. With the rapid progress in processing hardware, software, storage devices, and communication networks, these problems are being solved and video streaming applications are becoming increasingly popular.

In addition to the large storage, network bandwidth, and real-time constraints, with the proliferation of online multimedia content, it is also highly desirable that multimedia-streaming systems support effective and fast browsing. A key technique that enables fast and user friendly browsing of multimedia content is to provide full VCR functionality [72]. The set of effective VCR functionality includes forward, backward, stop (and return to the beginning), pause, step-forward, step-backward, fast-forward, fast-backward, and random access. This set of VCR functionality allows the users to have complete controls over the session presentation and is also useful for other applications such as video editing.

With the establishment of MPEG video coding standards, it is expected that many video sequences for streaming applications will be encoded in MPEG formats. However, the implementation of the full VCR functionality with the MPEG coded video is not a

trivial task. MPEG video compression is based on motion compensated predictive coding with an I-B-P-frame structure. The I-B-P-frame structure allows a straightforward realization of the forward-play function, but imposes several constraints on other trick modes such as random access, backward play, fast-forward play, and fast-backward play. As will be shown later, straightforward implementation of these functions requires much higher network bandwidth and decoder complexity compared to those required for the regular forward-play function.

With the I-B-P structure, to decode a P-frame, the previously encoded I/P-frames need to be decoded first. To decode a B-frame, both the I/P-frames before and after this B-frame need to be first decoded. To implement a backward-play function, a straightforward implementation is for the decoder to decode the whole group of picture (GOP), store all the decoded frames in a large buffer and play the decoded frames backward. However this will require a huge buffer (e.g., an N-frame buffer, if the GOP size is N) in the client machine to store the decoded frames which is not desirable. Another possibility is to decode the GOP up to the current frame to be displayed, and then go back to decode the GOP again up to the next frame to be displayed. This does not require the huge buffer but will require the client machine to operate in an extremely high speed (up to N times of the normal decoding speed) which is also not desirable. The problem soon becomes impractical when the GOP size is large.

Besides the problem with backward-play, fast-forward/backward and random-access also present difficulties. When a P/B-frame is requested, all the related previous P/I-frames need to be sent over the network and decoded by the decoder. This requires

the network to send all the related frames besides the actually requested frame at a much higher rate which can be many times of that required by the normal forward-play. When many clients request the trick-modes, it may result in much higher network traffic compared to the normal forward-play situation. It also requires high computational complexity in the client decoder to decode all these extra frames. It is possible to just send the I-frames for these trick-modes. However, if the applications use a very large GOP-size, or require high-precision in video-frame access, sending I-frames only may not be acceptable.

There are many different schemes to encode the MPEG video, depending on the desirable server/network/client complexity requirements. For example, the video can be encoded with all I-frames. This will result in the lowest complexity requirement for the client machines. However, it will require very large server storage and network bandwidth since the I-frames will result in high-bit-rates. Since the network bandwidth usually is the highest concern, we assume that the video is coded with all I-B-P frames that can achieve high compression ratios for the transport over a network with minimum bandwidth resources.

Some recent works have addressed the implementation of VCR functions for MPEG compressed video for streaming video applications [73][74][75][76]. References [73][74][75] address the problem of reverse-play of MPEG video streams, and reference [76] addresses the problem of fast-forward play. Chen *et al.* [73] described a method of transforming an MPEG I-B-P compressed bit-stream into a local I-B bit-stream by performing a P-to-I frame conversion to convert all the retrieved P-frames into I-frames

at the client, thereby breaking the inter-frame dependencies between the P-frames and the I-frames. After the frame conversion and frame reordering, the motion vector swapping approach developed in [74] can be used for the backward-play of the new I-B bit-stream. However, this approach requires higher decoder complexity to perform the P-to-I conversion and higher storage cost to store the bit-streams. Wee *et al.* [75] presented a method which divides the incoming I-B-P bit-stream into two parts: I-P frames and B-frames. A transcoder is then used to convert the I-P frames into another I-P bit-stream with a reversed frame order. A method of estimating the reverse motion vectors for the new I-P bit-stream based on the forward motion vectors of the original I-P bit-stream as described in [16] is used to reduce the computational complexity of this transcoding process. For B-frames, the motion vector swapping scheme proposed in [74] is used for the reverse-play. The transcoding process, however, still requires much computation and will cause drift due to the motion vector approximation [75]. None of the methods mentioned above fully address the problem of the extra network traffics and decoding complexity caused by the VCR functions such as fast-forward/backward and random-access. Omoigui *et al.* [76] investigated possible client-server time-compression implementations for fast-forward play and video browsing. The time-compression can be implemented by storing multiple pre-encoded bit-streams with different temporal resolutions and send a bit-stream with suitable temporal resolution according to the user's request. This approach does not introduce excessive network traffic but the speed-up granularity is limited by the number of pre-stored bit-streams.

In this chapter, we investigate effective techniques to implement the full VCR functionality in an MPEG video streaming system. We analyze the impacts of performing VCR trick-modes on the client decoder complexity and network traffics. We propose to use dual bit-streams at the server to resolve the problem of reverse-play. Based on the dual-bit-stream structure, we propose a novel frame-selection scheme at the server to minimize the required network bandwidth and the decoder complexity. This scheme determines the frames to stream over the network by switching between the two bit-streams based on a least-cost criterion. We present a drift-compensation scheme to eliminate the drift caused by the bit-stream switching. We also describe our implementation of an MPEG-4 video streaming system supporting the full VCR functionality.

The rest of this chapter is organized as follows. In Section 5.2, we discuss the impacts of random-access and fast-play operations on decoder complexity and network traffics. In Section 5.3, we describe our proposed scheme for supporting full VCR functionality with least network resource and decoding effort. Section 5.4 presents a drift-compensation scheme for the proposed least-cost bit-stream switching method. Finally, conclusions are given in Section 6.

5.2. Impacts of VCR Functionality on Decoder complexity and Network Traffic

A block diagram of an MPEG video streaming system is shown in Figure 35. The video streams are compressed using MPEG video coding standards and are stored in the

server. The clients can view the video while the video is being streamed over the network. In each client machine, a pre-load buffer is set up to smooth out the network delay jitter. In this paper we discuss the scenario that the video is streaming over the Internet and the full VCR functionality needs to be supported. It is assumed that the applications may use a large GOP size, or require relatively high precision in video-frame access.

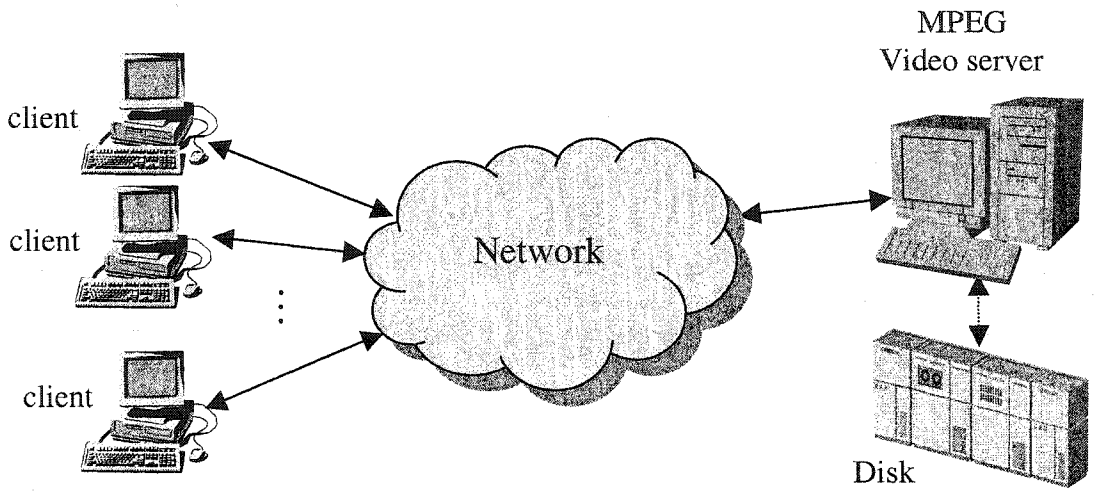


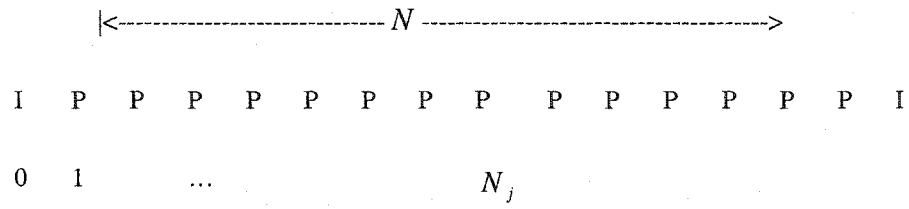
Figure 34. MPEG video streaming

In the following, we provide some analyses and simulation results to show the average number of frames needed to be sent through the network and decoded at the client decoder to support random-access and fast-forward play. Since the non-selected B-frames are not involved in decoding later frames and are not needed to be sent over the network or decoded by the decoder, for simplicity but without loss of generality, we focus the analyses on the cases that the bit-stream contains I- and P-frames only. The results can be easily extended to the I-B-P frame structure.

A. Random Access

In the random-access operation, the decoder requests a frame with an arbitrary distance from the current displayed frame. If the requested frame is an I-frame, the server side only needs to transmit this frame, and the decoder can decode it immediately. However, if the requested frame is a P-frame, the server needs to transmit all the P frames from the previous nearest I-frame to this requested frame.

Suppose all the GOPs in the bit-stream have the same length N, and frame N_j is the random-access point.



Then, in order to decode frame N_j, frames 0, 1,... $N_j - 1$ should also be sent from the server side. Assuming the random-access points are uniformly distributed, the average number of frames to be transmitted is $\bar{N}_{\text{trans}} = \frac{N+1}{2}$. For example, when $N = 14$, $\bar{N}_{\text{trans}} = 7.5$, meaning that an average of 7.5 frames should be transmitted over the network and decoded by the decoder for the requested frame in the random-access mode.

B. Fast-forward play

Suppose frame N_j is the starting point of the fast-forward operation, and k is the fast-forward speed-up factor (i.e. for k=6, only one out of 6 frames will be displayed). Since the next frame to be displayed is N_{j+k}, the server may send the frames $N_{j+1}\ N_{j+2} \ldots N_{j+k}$, so that k frames will be received by the client side to decode the frames $N_{j+1}\ N_{j+2} \ldots N_{j+k}$ (but just displays the frame N_{j+k}).

In fact, the server may not need to transmit so many frames. For example, consider the case:

```
          9                14 15 16 17 18 19
... P P P P P I P P P P P...
```

where frame 9 is the current displayed frame, and frame 15 is the next frame to be displayed under the fast-forward mode (k=6). Apparently, there is no need to send frames 10-13, since they are not needed for the decoding of frame 15. Therefore, the server can just send frames 14 and 15.

It is useful to derive a closed-form formula to show the impact of the fast-forward play on the decoding complexity and network traffics. One difficulty is, similar to the random-access operation, the start point of the fast-forward mode can be any frame in a GOP. However, it is reasonable to assume that the start point of a fast-forward operation is an I-frame, since we can always jump to the nearest I-frame first which will not cause unpleasant effect in viewing the video in most practical applications. Note that, with this assumption, after k/L GOPs, where L=gcd(k,N) stands for the greatest common devisor of

k and N, the frame to be displayed will again be an I-frame. Therefore, The decoding pattern will repeat every k/L GOPs (i.e., lcm(k,N) frames, where lcm(k,N) is the least common multiple of k and N). We can thus derive an analytical closed-form formula based on the periodicity. In the following, we divide different combinations of N and k into three classes and derive the closed-form formula respectively:

Case 1: $k > N$, $k \bmod N = 0$

In this case, all the P-frames are dropped, only the non-skipped I-frames are transmitted and decoded. No extra frames need to be transmitted for decoding the I-frames. Therefore $\bar{N}_{\text{trans}} = 1$.

Case 2: $k > N$, $k \bmod N \neq 0$

As mentioned above, the decoding pattern will repeat every k/L GOPs. During each period, there are N/L frames to be requested for display. For the i-th requested frame in each period ($i=0$ *to* $N/L - 1$), a total of $(i \times k) \bmod N + 1$ (where "mod" stands for the modular operation) frames need to be transmitted and decoded. The average number of frames to be transmitted and decoding for displaying one frame is

$$\begin{aligned} \bar{N}_{\text{trans}}(k,N) &= \frac{L}{N}\sum_{i=0}^{\frac{N}{L}-1}\left((i\times k)\bmod N+1\right) \\ &= \frac{L}{N}\sum_{i=0}^{\frac{N}{L}-1}\left(i\times L+1\right) \end{aligned} \tag{12}$$

Case 3: $2 \leq k \leq N\text{-}1$, $N \bmod k \neq 0$

In a GOP with an I-P structure, a P-frame needs not be sent only if all its following P-frames will not be displayed at the client decoder. Therefore, in the first GOP (assuming the start point is an I-frame), the number of frames needs not be transmitted is N mod k. Similarly, the number of the P-frames which need not be sent in the j-th GOP, where $1 \leq j < k/L$, is $(j \times N)$ mod k-1. Thus, the total number of frames that need not be transmitted in the k/L GOPs is

$$\begin{aligned} N_{\text{skip}}(k,N) &= \sum_{j=1}^{\frac{k}{L}-1} \left(\left(j \times N \right) \bmod k - 1 \right) \\ &= \sum_{j=1}^{\frac{k}{L}-1} \left(j \times L - 1 \right) \end{aligned} \tag{13}$$

If N and k are coprime (i.e., $L = 1$), the above equation becomes

$$N_{\text{skip}}(k) = \sum_{j=1}^{k-1} \left(j - 1 \right) \tag{14}$$

Note that, in the case that N and k are coprime, equation (14) holds for any start points (not necessarily an I-frame).

In case 3, the average number of frames that need to be transmitted and decoded for displaying a requested frame can be obtained by subtracting the non-transmitted frames from the total number of frames, and then dividing it by the total number of frames to be displayed. That is

$$\begin{aligned}\bar{N}_{\text{trans}}(k,N) &= \frac{\frac{k}{L}\times N - N_{\text{skip}}(k,N)}{\frac{k}{L}\times\frac{N}{k}} \\ &= k - \frac{L}{N}N_{skip}(k,N)\end{aligned} \tag{15}$$

In summary, the average number of frames need to be transmitted and decoded for a requested frame can be expressed in close-form as follows:

$$\bar{N}_{\text{trans}}(k,N) = \begin{cases} 1 & k{=}1 \text{ or } k \bmod N = 0 \\ k - \frac{L}{N}\sum_{i=1}^{\frac{k}{L}-1}(i\times L-1) & 2 \le k \le N-1 \\ \frac{L}{N}\sum_{i=0}^{\frac{N}{L}-1}(i\times L+1) & k > N,\ k \bmod N \neq 0 \end{cases} \tag{16}$$

Equation (17) suggests that, if N is relatively large compared to k, $\bar{N}_{\text{trans}}$ will grow almost linearly as k increases, thereby leading to a linear increase of the decoding complexity and the network traffics.

The above analyses can be extended to the case with B-frames. The main difference from the above analyses is that, to decode a B-frame, we only need to decode the related I/P-frames; the other B-frames do not need to be transmitted or decoded. Therefore, the number of frames needs to be transmitted and decoded for displaying one frame for GOPs with the general I-B-P structure is in general less than the I-P case. However, the analysis is very similar to the above. For simplicity of discussion, we

assume that $k < N$, and divide it into two cases. Again, we assume the start point is always an I-frame.

Case 1: k is a multiple of M (where M is the distance between the I-P or P-P frames)

In this case, all the B-frames need not be transmitted and decoded. Equation (17) can be applied by replacing k and N with k/M and N/M respectively.

Case 2: k is not a multiple of M

In this case, Equation (13) should be modified as follows.

$$N_{\text{skip}}(k,N) = \sum_{i=1}^{\frac{k}{L}-1} (i \times L - 1) + N_{\text{B\_skip}} - N_{\text{P\_decodeB}} \tag{17}$$

where $N_{\text{B\_skip}}$ represents the number of B-frames needs not be decoded, and $N_{\text{P\_decodeB}}$ represents the number of P-frames need to be transmitted and decoded for decoding the B-frames in those GOPs in which the last displayed frame is a B-frame. It is difficult to find closed-form representations for the second and third terms at the right hand side of the above equation. Computer simulations can be used to determine the numbers $N_{\text{B\_skip}}$ and $N_{\text{P\_decodeB}}$ for different combinations of N, M, and k.

Figure 36 shows the average number of frames needs to be sent and decoded for decoding a requested frame with respect to different speed-up factors in the fast-forward operation. The test MPEG bit-stream used for simulation is the "Mobile and Calendar" sequence with a length of 280 frames (20 GOPs in our example), which was encoded at 3 Mbps with a frame-rate of 30 fps with an I-P structure. The start points are randomly generated. Figure 37 depicts the average bit-rates required for sending the "Mobile and

Calendar" video stream with respect to different speed-up factors. From the above analysis, in the fast-forward/backward and random-access operations, the server needs to send several extra frames to the decoder to display one frame, thereby resulting in a heavy burden on the network (especially when the number of users is large) and increasing the decoder complexity.

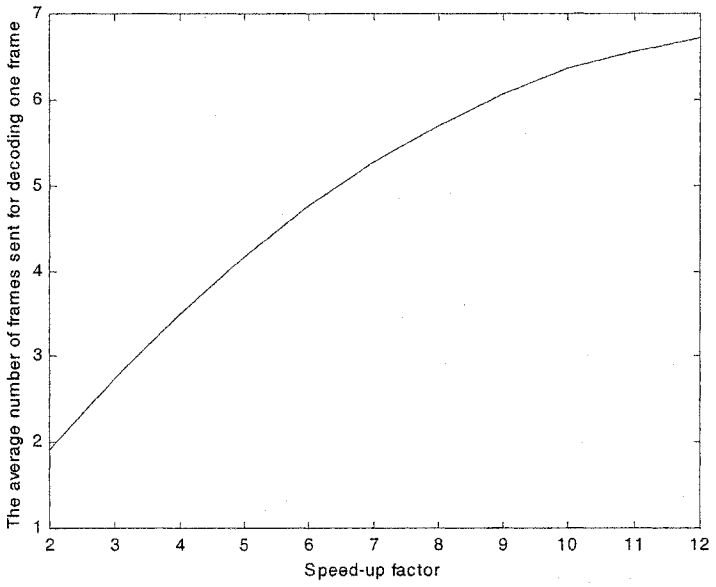


Figure 35. Average number of frames needs to be sent for decoding a frame with respect to different speed-up factors in fast-forward play.

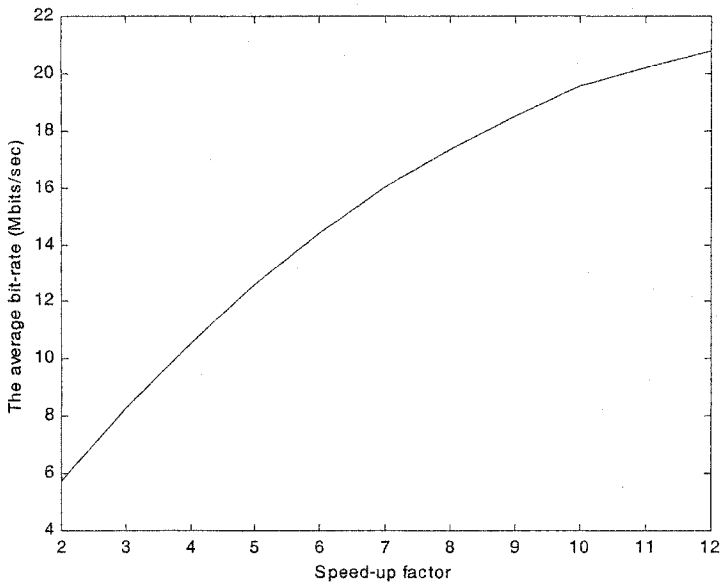


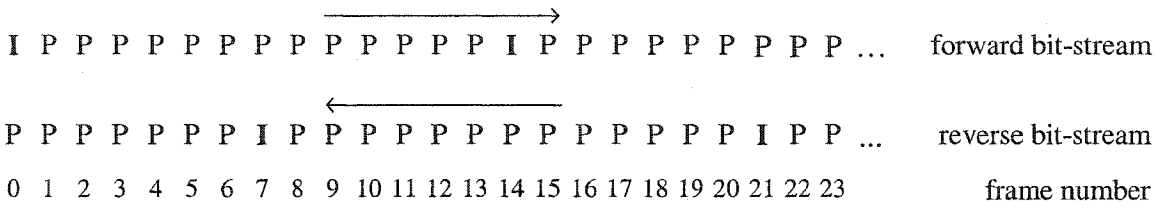
Figure 36. Average bit-rates for sending the "Mobile and Calendar" sequence over network with respect to different speed-up factors in fast-forward play.

5.3. Supporting Full VCR Functionality with Minimal Network bandwidth and Decoder Effort

5.3.1. Dual Bit-streams with Least-cost Frame Selection

To solve the problem of the backward-play operation, we propose to add a reverse-encoded bit-stream in the server, i.e., in the encoding process, after we finish the encoding and reach the last frame of the video sequence, we encode the video frames in the reverse order to generate a reverse-encoded bit-stream. If the server only has the forward bit-stream (i.e., the original sequence is unavailable), we can decode the forward bit-stream up to two GOPs each time in the reverse direction (i.e., from the last GOP to

the first GOP) then re-encode the video in the reverse order. The generation of the reverse bit-stream is done off-line. For simplicity of the presentation, in this paper, we use an example in which the video is coded in I/P-frames with a GOP size of 14 frames as shown below. The extension of our discussion to the case with the general I-B-P GOP structure is straightforward.



In the above diagram, we arrange the encoding so that the I-frames in the reverse bit-stream are interleaved between I-frames in the forward bit-stream. In this way, the required number of frames sent by the server and decoded by the decoder can be further reduced as will be explained later. Alternatively, the I-frames in both streams can be aligned to save storage since the two I-frames in the forward and reverse bit-streams are the same, and only need to be stored once. Two metadata files recording the location of the frames in each compressed bit-stream are also generated so that the server can switch from the forward-encoded bit-stream to the reverse-encoded bit-stream and vice versa easily. I-frames represent the points of access to decode the sequence from any arbitrary position. With the reverse-encoded bit-stream, when the client requests the backward-play mode, the server will stream the bits from the reverse-encoded bit-stream. Using this scheme, the complexity of the client machine and the required network bandwidth for the

backward-play mode can be minimized. The storage requirement of the server will be about doubled. However, this is usually much more desirable than to require a large network bandwidth and to increase the complexity of the client machine since the network bandwidth is more precious and there may be a large number of client machines in the streaming video applications. Since the encoding of the video is done off-line and can be automated, the extra time needed in producing the reverse encoded bit-stream is not an important concern.

To reduce the decoding complexity and the network traffics in the fast forward/backward and the random access modes, we propose a frame-selection scheme which minimizes a predefined "cost" using bit-stream switching. The cost can be the decoding effort at the client decoder or the traffic over the networks, or a combination of both. This is further explained as follows.

Let c_{R_C} stands for the cost of decoding the next requested P-frame from the current displayed frame, c_{R_FI} for the cost of decoding the next requested P-frame from the closest I-frame in the forward bit-stream, and c_{R_RI} for the cost of decoding the next requested frame from the closest I-frame of the reverse-encoded bit-stream. To minimize the number of frames sent to the decoder, the costs can be the distances from the possible reference frames to the next requested frame. To minimize the network traffic, the costs can be the numbers of bits from the possible reference frames required for decoding the next requested frame. The bit-rate calculation can be done simply by recording the number of bits used for each encoded frame in the metadata file in the pre-encoding

process, and summing up the bit-rates of those frames to be sent. In general, a larger number of frames to be sent implies heavier network load. However, it also depends on the numbers of I-, P-, and B-frames to be sent since the numbers of bits produced by these three types of frames vary greatly. It is also possible to use different weights to combine the two costs according to the channel condition and the client capability. Based on the current play-direction, the requested mode, and the costs c_{R_C}, c_{R_FI}, and c_{R_RI}, the reference frame to the next requested frame with the least cost will be chosen to initiate the decoding. This will also determine the selection of the next bit-stream and the decoding direction. This least-cost criterion will only be activated in the fast forward/backward and the random access modes to avoid frequent bit-stream switching in the normal forward/backward operations.

To illustrate the scheme, we use the example in Section 2 again, assuming that the previous mode was backward-playing and the requested mode is fast-backward with a speed-up factor 6 which needs to display a sequence of frame numbers 20, 14, 8, 2, ... For simplicity, in the following examples we use the minimum decoding distance criterion to illustrate the selection of the next reference frame and the effectiveness of the proposed method.

Frame No.		0	1	**2**	3	4	5	6	7	**8**	9	10	11	12	13	**14**	15	16	17	18	19	**20**	21	22	23	
F	->	**I**	P	P	P	P	P	P	P	P	P	P	P	P	P	**I**	P	P	P	P	P	P	P	P	P	...
R	<-	P	P	P	P	P	P	P	**I**	P	P	P	P	P	P	P	P	P	P	P	P	**I**	P	P	...	

The algorithm will operate as follows:

1. The current position is frame 20 which was decoded using the reverse bit-stream (*R*).
2. Frame 14 will be decoded from the forward bit-stream (*F*) directly since it is an I-frame.
3. Frame 8 will be decoded from frame 7 of the backward bit-stream, since the distance between frame 7 of the reverse bit-stream (an I-frame) and the requested frame (frame 8) is less than the distances between the requested frame and the current decoded frame (frame 14 of the reverse bit-stream), and the closest I-frame of the forward bit-stream (it's also frame 14). Note that, in this case, we use frame 7 of the reverse bit-stream (an I-frame) as an approximation of frame 7 of the forward bit-stream (a P-frame) to predict frame 8 of the forward bit-stream. This will cause some drift. However, in the fast-forward/backward modes, the drift is relatively insensitive to human eyes due to the fast change of the content displayed. Also, any I-frame in the play will terminate the drift. The drift problem will be further investigated in the next section.
4. Frame 2 will be decoded from frames 0 and 1, using the forward bit-stream, since the decoding effort from frame 0 of the forward bit-stream (an I-frame) is the minimum.

The bit-stream sent from the server will have the following form:

P	**I**	I	**P**	I	P	**P**	...	frame type
20	**14**	7	**8**	0	1	**2**	...	frame number
R	***F***	*R*	***F***	*F*	*F*	***F***	...	selected bit-stream

The frames indicated by the bold-face are those to be displayed at the client side. In this way, we only need to send and decode 6 frames. Without the minimum effort decoding scheme, we will need to send and decode 13 frames from the reverse bit-stream.

In the case of random access, frame skipping will be performed followed by normal forward-play. For example, the client requests random access to frame 22 when the current decoded frame is frame 3. With the proposed method using the minimum decoding distance criterion, the server streams the bit-stream as follows:

P	I	**P**	**P**	**P**	...	frame type
3	21	**22**	**23**	**24**	...	frame number
F	***R***	***F***	***F***	***F***	...	selected bit-stream

In this example, we only need to send and decode 2 frames to reach frame 22. Without our proposed least-cost scheme, it will require to send and decode 9 frames from frame 14 (an I-frame) using the forward bit-stream. Again, in this example, for frame 21, we use the I-frame in the reverse bit-stream to approximate the P-frame in the forward bit-stream. This will cause drift but the drift will only last a few frames within the GOP (a fraction of a second) since the video content will be refreshed by the I-frame in the next GOP. Thus it should not be a problem.

If the minimum decoding distance criterion is used (i.e., to minimize the number of frames sent to the decoder), the proposed scheme will guarantee that the maximum amount of decoding to access any frame in the sequence is less than $N/4$ frames if the I-frames in the forward and the reverse bit-streams are interleaved. In addition, no large

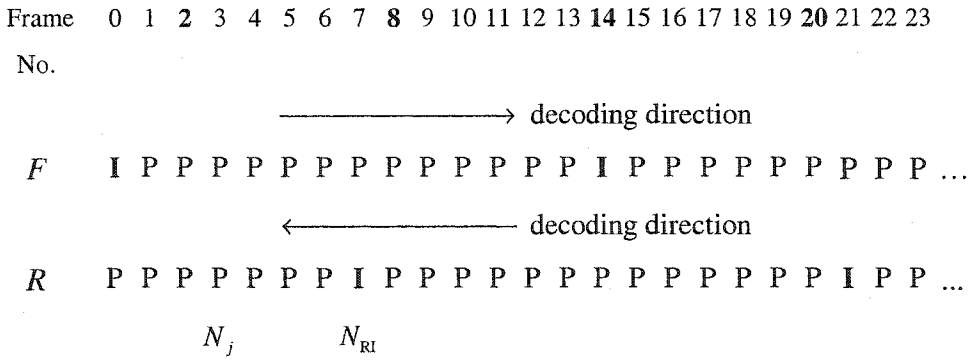
temporary buffer is required in the decoder. If the I-frames in the forward and the reverse bit-streams are aligned, the maximum amount of decoding to access any frame will be less then $N/2$ frames.

5.3.2. Dual Bit-streams with Least-cost Frame Selection

In the following, we analyze the performance of the proposed method using the minimum decoding distance criterion for the random-access and the fast-forward/backward modes.

A. Random Access

A typical structure of the system with dual bit-streams is shown below, where both the forward bit-stream and the reverse bit-stream have the same GOP structure with the length of N. As the analysis in Section 2, we label the frames in the GOP where the requested frame lies as 0, 1, 2, ..., N-1, frame N_j is the random access point. N_{RI} is the position of the I-frame in the reverse bit-stream.



In this case, the frames to be transmitted for decoding frame N_j will be decided by two distance measures, one is the distance from frame N_j to the nearest I frame in the

forward bit-stream, and the other distance is from N_j to the nearest I-frame in the reverse bit-stream.

Assume $N_j \in [0, N-1]$, N_{RI} is in the range of $[1, N-1]$. For simplicity but without loss of generality, we assume that N is even and N_{RI} is odd as in our previous example. We can observe that:

- The minimum number of total frames to be sent over the network is 1 (when $N_j = 0$ or $N_j = N_{\mathrm{RI}}$);
- The maximum number of total frames to be sent over the network is

$$\max\left(\frac{N_{\mathrm{RI}}-1}{2}+1, \frac{N-1-N_{\mathrm{RI}}}{2}+2\right);$$

- The average number of total frames to be sent over the network is:

$$\begin{aligned}\bar{N}_{\mathrm{trans}} &= 1+\sum_{i=0}^{\frac{N_{\mathrm{RI}}-1}{2}}\frac{i}{N}+\sum_{i=\frac{N_{\mathrm{RI}}+1}{2}}^{N_{\mathrm{RI}}}\frac{(N_{\mathrm{RI}}-i)}{N}+\sum_{i=N_{\mathrm{RI}}+1}^{\frac{N_{\mathrm{RI}}-1+N}{2}}\frac{(i-N_{\mathrm{RI}})}{N}+\sum_{i=\frac{N_{\mathrm{RI}}+1+N}{2}}^{N-1}\frac{(N-i)}{N}\\ &= 1+\frac{2N_{\mathrm{RI}}^2-2NN_{\mathrm{RI}}+N^2-2}{4N}\end{aligned}$$

By taking the derivative with respect to N_{RI}, we can find that when N_{RI} takes the odd number closest to $\frac{N}{2}$, $\bar{N}_{\mathrm{trans}}$ can take the minimum value of

$$\bar{N}_{\mathrm{trans}} = \begin{cases} 1+\frac{N}{8}-\frac{1}{2N} & (\frac{N}{2}=\text{odd}) \\ 1+\frac{N}{8} & (\frac{N}{2}=\text{even}) \end{cases}$$

For example, when $N = 14$, $N_{RI} = 7$, $\bar{N}_{trans} = 2.71$, meaning that an average of 2.71 total frames should be transmitted to decode one requested frame in the random access mode. Apparently, this is much better than the case in Section 2 (7.5 total frames without our scheme).

B. Fast forward-play

In the proposed method, when the speed-up factor k is larger than $N/4$, the server always can find an I-frame in one of the two bit-streams which has a shorter distance to the next displayed frame from the current displayed P-frame, since the distance for the nearest I-frame is guaranteed to be equal to or less than $N/4$. In this case the number of frames to be sent for displaying a requested frame will have a range of [1, $N/4+1$]. It is difficult to derive a closed-form formula to calculate the average number of frames transmitted and decoded for each requested frame for general N and k cases with any start points. For simplicity of analysis, in the following, we assume the start point is an I-frame in the forward bit-stream and only consider the case that $k > N/4$.

Since the start point is an I-frame, the requested frames will again become I-frames every k/L GOPs, meaning that the decoding pattern will repeat every k/L GOPs. The distance between the i-th requested frame (i=0,1,...) and its proceeding I-frame is $i \times k$ mod N, which is a multiple of L. No distances will be the same in a period, otherwise we can find a shorter period which is a contradiction. Therefore we can conclude that, in a period of k/L GOPs, there are a total of N/L requested frames with equally-spaced distances, say 0, L, $2\times L$, ..., $(N/L-1)\times L$ frames away from their nearest proceeding I-

frames. Based on this property, we can derive closed-form formulas to calculate the average number of frames transmitted for decoding a requested frame. We divide the analysis into two classes:

Case 1: N is even

In this case, every $N/2$ frames there is an I-frame (in either the forward or the backward bit-streams, when the I-frames are interleaved) which can be used as an anchor frame to initiate the decoding of the requested frames. The average number of frames transmitted for decoding a requested frame is

$$\begin{aligned}\bar{N}_{\text{trans}} &= 1+\frac{1}{\frac{N}{L}}\left(\sum_{i=0}^{\left\lfloor\frac{N}{4L}\right\rfloor} i\times L+\sum_{i=\left\lfloor\frac{N}{4L}\right\rfloor+1}^{\left\lfloor\frac{N}{2L}\right\rfloor}\left(\frac{N}{2}-i\times L\right)+\sum_{i=\left\lfloor\frac{N}{2L}\right\rfloor+1}^{\left\lfloor\frac{3N}{4L}\right\rfloor}\left(i\times L-\frac{N}{2}\right)+\sum_{i=\left\lfloor\frac{3N}{4L}\right\rfloor+1}^{\frac{N}{L}-1}\left(N-i\times L\right)\right)\\ &= 1+\frac{2L}{N}\left(\sum_{i=0}^{\left\lfloor\frac{N}{4L}\right\rfloor} i\times L+\sum_{i=\left\lfloor\frac{N}{4L}\right\rfloor+1}^{\left\lfloor\frac{N}{2L}\right\rfloor}\left(\frac{N}{2}-i\times L\right)\right)\end{aligned} \tag{18}$$

where $\lfloor x\rfloor$ represent the largest integer smaller than X. It is interesting to note that, when N and k are coprime (i.e., $L = 1$), $\bar{N}_{\text{trans}}$ is only a function of N regardless of the values of k. The above equation becomes

$$\begin{aligned}\bar{N}_{\text{trans}} &= 1+\frac{2}{N}\left(\sum_{i=0}^{\left\lfloor\frac{N}{4}\right\rfloor} i+\sum_{i=\left\lfloor\frac{N}{4}\right\rfloor+1}^{\frac{N}{2}}\left(\frac{N}{2}-i\right)\right)\\ &= \begin{cases} 1+\frac{N}{8} & \frac{N}{2} \text{ is even}\\ 1+\frac{N}{8}-\frac{1}{2N} & \frac{N}{2} \text{ is odd}\end{cases}\end{aligned} \tag{19}$$

In fact, for the cases that N and k are not coprime, the results of (18) and (19) are still very close. Therefore, the simple formula in (19) can be applied in most of the cases that N is even.

Case 2: N is odd

Similar to the above derivation, we can obtain the following formula

$$\bar{N}_{\text{trans}} = 1 + \frac{1}{\frac{N}{L}} \left(\sum_{i=0}^{\left\lfloor \frac{N-1}{4L} \right\rfloor} i \times L + \sum_{i=\left\lfloor \frac{N-1}{4L} \right\rfloor + 1}^{\left\lfloor \frac{N-1}{2L} \right\rfloor} \left(\frac{N-1}{2} - i \times L \right) + \sum_{i=\left\lfloor \frac{N-1}{2L} \right\rfloor + 1}^{\left\lfloor \frac{3(N-1)}{4L} \right\rfloor} \left(i \times L - \frac{N-1}{2} \right) + \sum_{i=\left\lfloor \frac{3(N-1)}{4L} \right\rfloor + 1}^{\frac{N}{L}-1} \left(N - i \times L \right) \right) \quad (20)$$

When N and k are coprime, the above equation becomes

$$\bar{N}_{\text{trans}} = 1 + \frac{N}{8} - \frac{1}{8N} \quad (21)$$

We have simulated the situation of the I-P structure for N = 14 with a number of randomly generated start points. Two bit-streams generated by forward and reverse encoding the 280-frame "Mobile and Calendar" test sequence at 3 Mbps with the frame rate of 30 fps with an I-P structure are used for the simulation. Figure 38 shows the comparison of the average number of the frames transmitted to the decoder for decoding a requested frame with and without the proposed dual-bit-stream least-cost method with respect to different speed-up factors in the fast-forward operation. The simulation result is very close to the value 2.71 calculated by using (19) with N = 14 when the speed-up factor $k > N/4$. The fast-backward play case will also have similar result. Figure 39 depicts the comparison of the average bit-rates required to send the video stream with respect to

different speed-up factors. Note that, with the proposed method, when the speed-up factor reaches around $N/4$ (e.g., 3.5 in our example), the decoding complexity and the network traffic will not continue to grow even when the speed-up factor gets higher. Compared to the results in Section 2, it is obvious that the proposed method can achieve significant performance improvement in terms of the decoder complexity and the network traffic load. When the speed-up factor $k \geq N/4$, the proposed method guarantees a nearly constant decoding and network traffic cost.

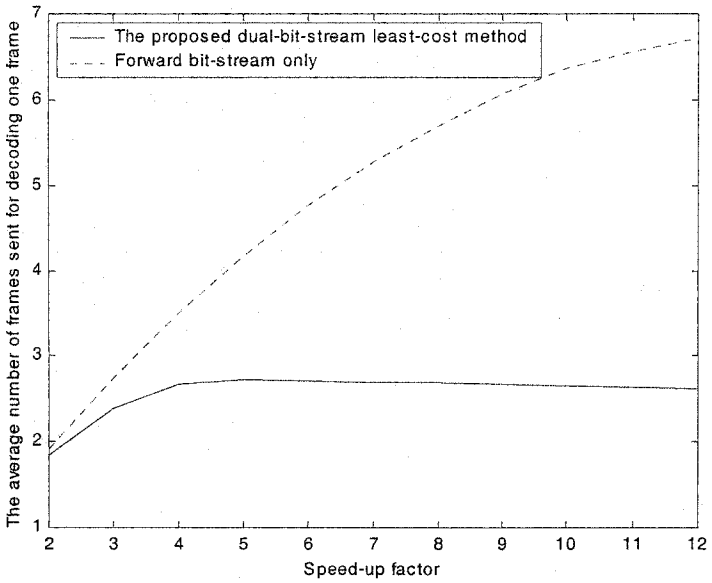


Figure 37. Estimated number of frames to be sent for decoding a frame using the proposed method with respect to different speed-up factors.

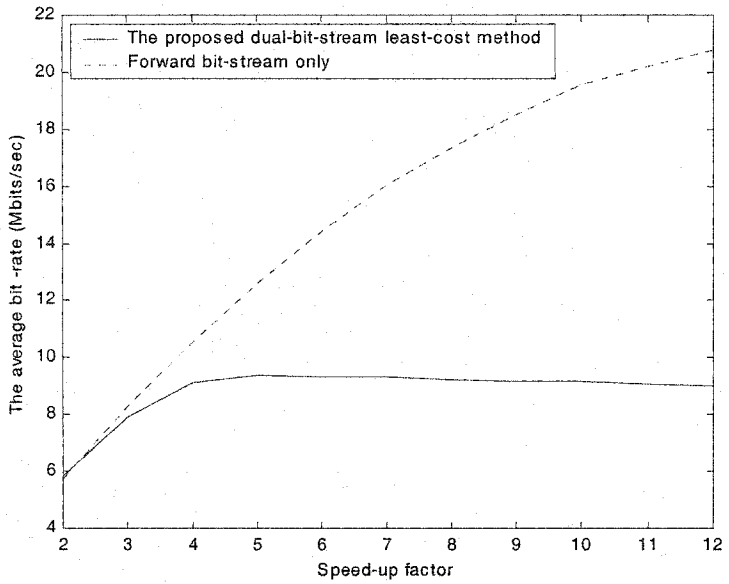


Figure 38. Average bit-rates to send the "Mobile and Calendar" sequence over network using the proposed method with respect to different speed-up factors.

5.4. Drift Compensation

As mentioned above, in the proposed scheme, I- or P-frames of one bit-stream may be used to approximate P-frames of the other bit-stream. This approximation, however, will lead to frame mismatch and thus cause drift when the approximated frames are used as the reference frames to predict the following P/B-frames as illustrated in Figure 39, where the "Mobile & Calendar" sequence is encoded at a fixed quantization scale (Q=16) and at a fixed bit-rate (3 Mbps) respectively. The GOP size is 14 and the speed-up factor is 6. As shown in Figure 6, when the server performs an I-to-P or a P-to-P approximation by using the proposed bit-stream switching, there is a PSNR drop. Figure 39 suggests that the drift caused by the bit-stream switching can be as large as 2.5 dB and

will last until the next I-frame. However, the subjective degradation observed is not significant, since the fast display speed in the fast forward/backward modes will mask most of the spatial distortions.

In the random access mode, the drift will only last a few frames within a GOP, thus will not cause serious degradation. In the fast forward/backward mode, the drift is relatively insensitive to human eyes due to the fast changes of the content displayed. However, in some applications, it may still be desirable to prevent the drift. The drift problem can be resolved by adding two bit-streams consist of all P-frames for the drift-compensated bit-stream switching. This is further explained using the following example:

Frame No.	0	1	**2**	3	4	5	6	7	**8**	9	10	11	12	13	**14**	15	16	17	18	19	**20**	21	22	23	
															→ decoding direction										
D^{FR}	**P**	P	P	P	P	P	P	P	P	P	P	P	P	P	**P**	P	P	P	P	P	P	P	P	P	
F	**I**	P	P	P	P	P	P	P	P	P	P	P	P	P	**I**	P	P	P	P	P	P	P	P	P	...
															← decoding direction										
R	P	P	P	P	P	P	P	**I**	P	P	P	P	P	P	P	P	P	P	P	P	**I**	P	P	...	
D^{RF}	P	P	P	P	P	P	P	**P**	P	P	P	P	P	P	P	P	P	P	P	P	**P**	P	P		

where D^{FR} is a bit-stream used for switching from the I- or P-frames of the forward bit-stream to the P-frames of the reverse bit-stream, while D^{RF} is used for switching from the I- or P-frames of the reverse bit-stream to the P-frames of the forward bit-stream. The bit-stream D^{FR} is obtained as follow:,

$$D_n^{FR} = \text{Pred}(F_n, R_{n-1}) \tag{22}$$

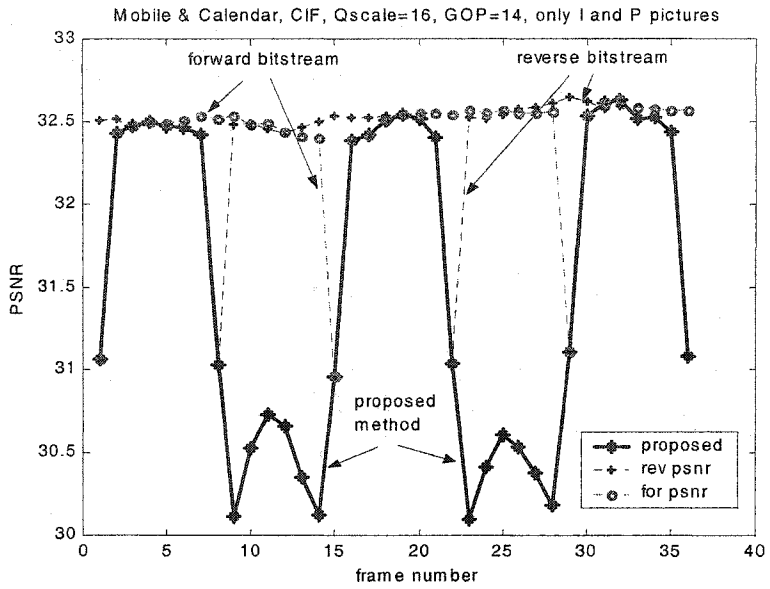
and

$$D_n^{RF} = \text{Pred}(R_n, F_{n+1}) \tag{23}$$

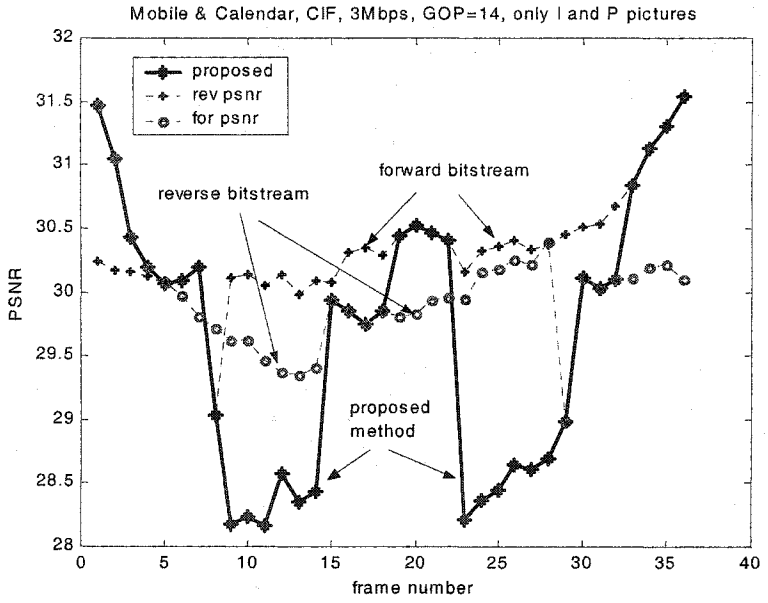
where Pred(A,B) represents an inter-frame prediction process that frame B is predicted from the reference frame A. When performing the bit-stream switching, the correctly predicted frame is used for switching between the forward and the reverse bit-streams. For example, if the bit-stream is switched from F_n (an I- or P-frame) to R_{n-1} (a P-frame), then the server will send the frames as ... F_n, D_n^{FR}, R_{n-2} ..., instead of sending ... F_n, R_{n-1}, R_{n-2}, ... With the two drift correction bit-streams, the proposed method will generate a bit-stream for the fast-reverse example in Section 5.3.1 as follows:

P	**I**	I	**P**	I	P	**P**	...	frame type
20	**14**	7	**8**	0	1	**2**	...	frame number
R	***F***	***R***	***D*^{RF}**	***F***	***F***	***F***	...	selected bit-stream

Since D^{RF} is the encoded based on the decoded frames from the forward and the reverse bit-streams, the drift can be compensated very well. If the prediction errors of the drift-compensated predictive frames in D^{RF} and D^{FR} are losslessly encoded, there will be no drift. Otherwise, there will be small drift. The drift will depend on the quantization step-size used in the encoding. A finer quantizer will lead to lower drift, while increasing the storage for the drift compensation bit-streams. Since the encoding process to obtain all the bit-streams is done off-line in streaming video applications, the encoding complexity is not a major concern.



(a)



(b)

Figure 39. PSNR comparison of the forward bit-stream, the reverse bit-stream, and the bit-stream generated using the proposed method in the fast-forward mode for the "Mobil and Calendar" sequence. (a) The sequence is quantized at Q=16; (b) the sequence is encoded at 3 Mbps.

It should be noted that if the I-frames of the two bit-streams are interleaved, and the speed-up factor is high enough (e.g., the frame skipping distance ≥ *N*/4), in the proposed method, only replacing P-frames with I-frames will be sufficient because we always can find an I-frame in one of the two bit-streams which has shorter distance to the next requested frame than the current decoded P-frame. In this case, we only need to store the drift compensation frames for all the I-frames of both bit-streams. In the fast-forward/backward operations with small speed-up factors (e.g., 2 or 3), however, the proposed least-cost scheme has limited gain on the decoding complexity and the network traffics as shown in Figures 4 and 5. Thus, a possible low-complexity solution for the fast-forward/reverse play is:

If $k < N/4$

Use dual bit-streams without performing bit-stream switching.

else

Use bit-stream switching with I -> P drift-compensation only.

Using this modified scheme, only the drift-compensations for the I -> P frames need to be created, thus the storage cost for the drift compensation frames can be reduced drastically without significant performance sacrifice in typical MPEG applications.

5.5. Summary

In this chapter, we discussed issues in implementing an MPEG video streaming system with full VCR functionality. We showed that when the users request reverse-

play, fast-forward/reverse-play, or random-access, it may result in much higher network traffic than the normal-play mode. These trick-modes may also require high client machine complexity. We proposed to use a reverse-encoded bit-stream to simplify the client terminal complexity while maintaining the low network bandwidth requirement. We proposed a minimum-cost frame-selection scheme which can minimize the number of frames needed to be sent over the network and to be decoded. We proposed a drift-compensation scheme to limit the drift. We also described our implementation of an MPEG-4 video streaming system. We showed that with our proposed scheme, an MPEG-4 video streaming system with full VCR functionality can be implemented to minimize the required network bandwidth and decoder complexity.

Chapter 6. Concluding Remarks

The transport of both live encoded video and pre-encoded stored video is an important part of real-time multimedia services. Our challenge is to develop solutions to overcome the problems caused by the heterogeneous transport environment and to provide user interactive capability. Scalable video coding offers the capability of dynamically adapting to the heterogeneous network conditions. The MPEG-4 FGS is an efficient scheme to provide the mentioned scalability with fine granular quality improvement. To improve the video quality of the MPEG-4 FGS bit-stream for video streaming applications, we have presented several new techniques. We also developed a novel method to offer flexible user-interactive capability for MPEG video streaming applications. In this chapter, we first summarize the major contributions of this dissertation, and then provide suggestions for future research.

6.1. Summary of Major Contributions

After introducing the concept and applications of the video transport over the Internet in Chapter 1, we give an overview of video coding in Chapter 2. We point out the two major research issues where we focus on: the first one is the MPEG-4 FGS enhancement layer truncation and transportation, and the second one is how to provide full VCR functionality to the end user.

In Chapter 3, we point out that current MPEG-4 FGS enhancement layer truncation schemes are not able to enhance the whole frame uniformly, which leads to

intra-frame quality variation. In order to solve this problem, we present our MPEG-4 FGS enhancement layer truncation scheme, where the available bit budget for the last enhancement layer to be truncated is redistributed throughout the whole frame area to raise the quality of different parts in the frame uniformly. An operational rate-distortion optimization scheme based on the Lagrange Multiplier (LM) algorithm is further adopted to improve the visual quality and reduce the intra-frame quality variation. Compared to the straightforward truncation, our approach reduces the intra-frame quality variation significantly, and the decoded visual quality in terms of PSNR is also improved.

In Chapter 4, we consider the issue of how to improve the robustness of the FGS bit-stream delivery in a congested network environment. We propose to utilize the multi-path features of today's networks to transmit the split version of the FGS enhancement layers with high bit-rates, so as to overcome the instantaneous network congestions and large bandwidth requirement for enhancement layers. We also propose techniques to split the high enhancement layers to reduce their rates for the multi-path transport. We initially split the original enhancement layer blocks by evenly allocating the bits in every block into different channels. Rate-distortion and other optimization schemes are then carried out on each description, so that the split coding efficiency and its robustness can be improved. Compared to the single-path transport approach, we can achieve better decoded visual quality.

In Chapter 5, we study the problem to provide full VCR functionality for the MPEG video streaming application. We point out that efficiently realizing the functions of random-access, fast-forward, and fast-backward play of the video in the compressed

domain is not a trivial task. We propose to add a reverse-encoded bit-stream and use a minimum-cost frame-selection scheme to minimize the number of frames to be sent over the network and to be decoded. Drift-compensation issue is also taken into consideration. With this proposed scheme, an MPEG video streaming system with full VCR functionality can be implemented to minimize the required network bandwidth and decoder complexity.

In summary, we have presented several new techniques for the streaming video applications. Specifically, for the transportation of the MPEG-4 FGS bit-streams, we provide the schemes that can help to improve the visual quality when the streams need to be truncated. We also propose to use multi-paths to transmit the split enhancement layers so as to overcome the instantaneous network congestions and improve the transport quality. For the user interactive issue, we propose a solution to provide full VCR functionality in the compressed domain. The proposed techniques are practical, and the results obtained are promising.

6.2. Suggestions for Future Research

Our enhancement layer truncation scheme is discussed in chapter 3, where the rate-distortion optimization is done within each enhancement layer frame with the given bit-budget. The performance of the proposed scheme can be improved by combining the approach to solve the inter-frame quality variation, such as [43]. The key idea is that, by considering several consecutive frames together, we can allocate the bit-budget among these frames according to their complexities, and the frames with larger intra-frame

quality variations will be assigned with more bits, so that the intra-frame quality variation will be uniform from frame to frame. After the frame level bit-allocation, we can continue the process described in chapter 3 for the optimization. However, we will need to try a better measurement of the frame complexity, so that we can allocate proper bit-budget to each frame.

In our multi-path MPEG-4 FGS enhancement layer splitting scheme presented in chapter 4, we consider that the two available channels are symmetric, i.e., having a similar bandwidth. Then, we apply the proposed layer splitting methods, which generate balanced descriptions of the original enhancement layers. We did not consider how the channel condition, e.g., packet loss rate, delay jitter, affects the layer splitting. However, in real application scenarios, the obtained multi-paths may have different bandwidths and different channel characteristics. It would be interesting to consider the conditions in both channels jointly. How to provide a splitting scheme with better quality and less computation burden for these general conditions needs to be further studied in the future.

Another topic we could try in the future is to extend the scalability concept to the H.264 coding standard. The H.264 video coding standard provides outstanding coding performance compared with other existing video coding standards. However, the complexity of the H.264 video codec increases a lot, and the standard itself does not support any scalability feature. For the decoders with limited computational capacity and memories, such as a media processor, it may not be able to achieve the real-time decoding for the bit-streams with many motion compensation modes, many required reference frames, and many intra-prediction modes. Products based on media processors

are more competitive if more functions run concurrently on the same processor. Scalable media processing [78] is proposed to achieve such an advantage by operating the decoding process at multiple complexity levels to adapt to different load situations. The decoding scalability discussed in [78] is achieved by discarding certain DCT coefficients out of all the 64 DCT coefficients in a coded block. However, in H.264, 4x4 integer transform is applied instead of the 8x8 DCT transform. The maximum number of coefficients in a coding unit is only 16, thus we will have less space in achieving the scalability by discarding the coefficients in the transformed domain. We can also achieve the scalable processing by discarding the motion compensation accuracy, (e.g., use full pixel to approximate the ½ pixel motion compensation accuracy). However, this will lead to severe error propagation to the areas in its following frames that will use the current area as motion compensation reference block. A possible solution is that how to organize the coding bit-stream on the encoder side so that it is easier for the decoder to discard the content to achieve the flexible computational scalability. However, we need further study on this topic.

List of References

[1] Microsoft Windows Media, Microsoft Corporation Inc., http://www.microsoft.com/windows/windowsmedia/.

[2] Apple QuickTime Player, Apple Corporation Inc., http://www.apple.com/quictime/.

[3] Real Networks RealPlayer, http://www.real.com/.

[4] Relay Networks ReplayTV, http://www.replay.com/.

[5] TiVo Inc., http://www.tivo.com/.

[6] D. Wu, Y. Hou, and Y. Zhang, "Transporting real-time video over the Internet: challenges and approaches," *Proc. IEEE*, Vol.88, No. 12, Dec. 2000, pp.1855-1877.

[7] S. Wenger, "H.264/AVC over IP," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 13, No. 7, July 2003, pp. 645-656.

[8] J. Postel and J.K. Reynolds, "File Transfer protocol," *RFC 959*, Internet Engineering Task Force, 1985.

[9] R. Fielding, J. Gettys, J. Mogul, H, Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol-HTTP/1.1," *RFC 2616*, Internet Engineering Task Force, 1999

[10] M. S. Chen, D. D. Kandlur, "Downloading and stream conversion: supporting interactive playout of videos in a client station," *Second Int. IEEE Conf. Multimedia Computing and Systems*, pp. 73-80, Washington, 1995.

[11] T. D.C. Little and D. Venkatesh, "Prospects for interactive video-on-demand," *IEEE Multimedia*, vol. 13, pp. 14-24, Aug. 1994.

[12] D. Wu, Y. Hou, W. Zhu, Y. Zhang, and J. Peha, "Streaming Video over the Internet: Approaches and Directions," *IEEE Trans. On Circuits and System for Video Technology*, Vol. 11, No. 3, March 2001, pp. 282-300.

[13] R. Braden, D. Clark and S. Shenker, "Integrated services in the Internet architecture: an overview," *RFC 1633*, Internet Engineering Task Force, 1994.

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," *RFC 2475*, Internet Engineering Task Force, 1998.

[15] T. Jiang, E. W. Zegura, and M. Ammar, "Inter-receiver fair multicast communication over the Internet," *Proceedings of NOSSDAV*, June 1999.

[16] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol.11, No.3, March 2001, pp301-317.

[17] J. Xin, "Improved standard-conforming video transcoding techniques," *Ph.D. dissertation, University of Washington*, 2002.

[18] A. Vetro, C. Christopoulos, and H. Sun, "Video Transcoding Architectures and Techniques: An overview," *IEEE Signal Processing Magazine*, March 2003, pp18-29.

[19] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG bitstreams," *Signal Processing: Image Communication*, Vol. 8, No. 6, Sep. 1996, pp. 481-500.

[20] R. Mohan, J.R. Smith, and C.-S. Li, "Adapting multimedia internet content for universal access," *IEEE Transactions on Multimedia*, Vol. 1, No. 1, March 1999.

[21] ISO/IEC CD 15938-5, "Information technology -- Multimedia content description interface -- Part 5: Multimedia description schemes," Edition 1, 2001. (MPEG-7 MDS).

[22] P. M. Kuhn, T. Suzuki, A. Vetro, "MPEG-7 transcoding hints for reduced complexity and improved quality," *International Packet Video Workshop 2001*, Kyongju, Korea.

[23] Y.Wang, and S. Lin, "Erorr-Resilient Video Coding Using Multiple Description Motion Compensation," *IEEE Trans. on Circtuis and Systems for Video Technology*, Vol. 12, No. 6, June 2002, pp 438-452.

[24] V. Goyal, "Multiple Description Coding: Compression Meets the Network," *IEEE Signal Processing Magazine*, Sept. 2001, pp 74-93.

[25] Coding of Audio-Visual Objects -- Part 2 Visual -- Amendment 2: Streaming Video Profiles, ISO/IEC 14496-2:2001/Amd 2:2002, 2002.

[26] W. Li, "Bit-Plane Coding of DCT Coefficients for Fine Granularity Scalability," ISO/IEC JTC1/SC29/WG11, MPEG98/M3989, Oct. 1998.

[27] P. A. Chou, H. J. Wang, and V. N. Padmanabhan, "Layered multiple description coding," *Packet Video Workshop*, Nantes, France, April 2003.

[28] *ISO/IEC 11172-2*, "Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5Mbit/s – Part 2: Video," Edition 1, 1993.

[29] *ISO/IEC 13818-2*, "Information technology – Generic coding of moving pictures and associated audio information: Video," Edition 2, 2000.

[30] *ISO/IEC 14496-2*, "Information technology – Coding of audio-visual objects – Part 2: Visual," Edition 1, 1999.

[31] *ITU-T Recommendation H.261*, "Video codec for audiovisual services at p × 64kbit/s," Mar 1993.

[32] *ITU-T Recommendation H.263*, "Video coding for low bit rate communication," Feb 1998.

[33] "Text of Final Committee Draft of Joint Video Specification," (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC), *ISO/IEC JTC1/SC29/WG11*, July 2002, Klagenfurt, AT.

[34] N. Ahmed, T. Natarajan, and K.R. Rao, "Discrete Cosine Transforms," *IEEE Trans. Computers*, C-23: 90-93, 1974.

[35] B. Girod, "The efficiency of motion-compensating prediction for hybrid coding of video sequences," *IEEE Journal on Selected Areas in Communications*, Vol. SAC-5, pp. 1140-1154, Aug. 1987.

[36] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 13, No. 7, July 2003, pp 560-576.

[37] N. Shacham, "Multipoint communication by hierarchically encoded data," in *Proc. IEEE INFOCOM '92*, May 1992, pp. 2107-2114.

[38] B. Girod, U. Horn, and B. Belzer, "Scalable video coding with multiscale motion compensation and unequal error protection," in *Proc. Symp. MultimediaCommunications and Video Coding*, New York, Oct. 1995, pp.475–482.

[39] Chapter 8 "Layered Coding," in "Compressed Video over Networks."

[40] Rama Kalluri, "Single-Loop Motion-Compensated based Fine-Granular Scalability (MC-FGS), " *MPEG2001/M6831*, Sydney, July, 2001.

[41] F. Wu, S. Li, and Y. Zhang, "A Framework for Efficient Fine Granularity Scalable Video Coding," *IEEE Trans. on Circuits and System for Video Technology*, Vol. 11, No. 3, March 2001, pp 332-344.

[42] L. Zhao, Q. Wang, S. Yang and Y. Zhong, "A Content-based Selective Enhancement Layer Dropping Algorithm for FGS Streaming Using Nearest Feather Line Method," *Visual Communications and Image Processing 2002*, Proceedings of SPIE, Vol. 4671, pp. 242-249.

[43] X.Zhang, A. Vetro, Y. Shi and H. Sun, "Constant Quality Constrained Rate Allocation for FGS Video Coded Bitstreams," *Visual Communications and Image Processing 2002*, Proceedings of SPIE, Vol. 4671, pp. 817-827.

[44] L. Zhao, J. Kim and C. Kuo, "MPEG-4 FGS Video Streaming with Constant-Quality Rate Control and Differentiated Forwarding," *Visual Communications and Image Processing 2002*, Proceedings of SPIE, Vol. 4671.

[45] W. Cheong, K. Kim, G. Park, Y. Lim, Y. Lee and J. Kim, "FGS coding scheme with arbitrary water ring scan order," *ISO/IEC JTC1/SC29/WG11, MPEG 2001/m7442*, Sydney, July, 2001.

[46] C. Lim and T. Tan, "Macroblock reordering for FGS," *ISO/IEC JTC1/SC29/WG11, MPEG 2000/m5759*, March, 2000.

[47] J. Zhou, H. Shao, C. Shen and M.T. Sun, "Multi-path Transport of FGS Video," *IEEE International Conference on Multimedia and Expo*, Baltimore, MD, 2003.

[48] J. Zhou, H. Shao, C. Shen and M.T. Sun, "FGS Enhancement Layer Truncation with Minimized Intra-Frame Quality Variation," *Packet Video* 2003.

[49] Y. Wang and Q. Zhu, "Error Control and Concealment for Video Communication: A Review," *Proceedings of the IEEE*, vol. 86, No. 5, May 1998, pp974-997

[50] Y. Wang, S. Wenger, et. al, "Error Resilient Video Coding Techniques," *IEEE Signal Processing Magazine*, July 2000, pp61-82.

[51] G. Wang, Q. Zhang, W. Zhu and Y. Zhang, "Channel-Adaptive Error Control for Scalable Video over Wireless Channel," *The 7th International Workshop on Mobile Multimedia Communications (MoMuC) 2000*, Oct., 2000, Japan.

[52] M. Van der Schaar and H. Radha, "Unequal packet Loss Resilience for Fine Granularity Scalability Video," *IEEE Trans. on Multimedia*, Vol. 3, No. 4, Dec 2001, pp. 381-394.

[53] Q. Wang, Z. Xiong, F. Wu and S. Li, "Optimal Rate Allocation for Progressive Fine Granularity Scalable Video Coding," *IEEE Signal Processing Letters*, Vol. 9, No. 2, Feb. 2002, pp 33-39.

[54] X. Zhang, A. Vetro, Y, Shi, and H. Sun, "Constant Quality Constrained Rate Allocation for FGS Video," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 13, No. 2, Feb. 2003, pp 121-130.

[55] H. Cheng, X. Zhang, Y. Shi, A. Vetro and H. Sun, "Rate Allocation for FGS Coded Video Using Composite R-D Analysis," *IEEE International Conference on Multimedia and Expo, Baltimore*, MD, 2003.

[56] M. van der Schaar and H. Radha, "A Hybrid Temporal-SNR Fine Granular Scalability for Internet Video," *IEEE Trans. on Circuits and System for Video Technology*, Vol. 11, No. 3, March 2001, pp. 318-331.

[57] R. Yan, F. Wu, S. Li and Y. Zhang, "Macroblock-based Progressive Fine Granularity Spatial Scalability (mb-PFGSS)," *ISO/IEC JTC1/SC29/WG11, MPEG2001/M7112*, March 2001.

[58] A. Ortega and K. Ramchandran, "Rate-distortion Methods for Image and Video Compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23-50, Nov. 1998.

[59] G. Sullivan and T. Wiegand, "Rate-distortion Optimization for Video Compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74-90, Nov. 1998.

[60] Y. Shoham, and A. Gersho, "Efficient Bit Allocation for an Arbitrary Set of Quantizers," *IEEE Trans. on Acoustic, Speech and Signal Processing*, Vol. 36, No. 9, Sept. 1988, pp 1445 – 1453.

[61] R. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Transactions on Image Processing*, Vol. 2, No. 2, pp. 160-175, Apr. 1993.

[62] L. Zhao, W. Qi, S. Li, S. Yang and H. Zhang, "A New Content-based Shot Retrieval Approach: Key-frame Extraction based Nearest Feature Line (NFL) Classification," *ACM Multimedia Information Retrieval 2000*, Los Angeles, Oct. 30-Nov. 4, 2000.

[63] Israel Cidon , Raphael Rom , Yuval Shavitt, "Analysis of Multi-path Routing," *IEEE/ACM transaction on Networking*, Vol. 7, No. 6, pp. 885-896, December 1999.

[64] P. Pham, S. Perreau, "Multi-path routing protocol with load balancing policy in mobile ad hoc network," *4th International Workshop on Mobile and Wireless Communications Network*, 2002.

[65] J. Byers, J. Considine, M. Mitzenmacher and S. Rost, "Informed Content Delivery Across Adaptive Overlay Networks," *ACM Sigcomm* 2003.

[66] H. Zheng and D. Samardzija, "Wireless Video Performance Through BLAST Testbed," *VTC 2001*, pp141-146.

[67] A. R. Reibman, Yao Wang, et. al, "Transmission of Multiple Description and Layered Video over an EGPRS Wireless Network," *International Conference on Image Processing, 2000*, pp136-139.

[68] N. Kamaci, Y. Altunbasak, R. M. Mersereau, "Multiple description coding with multiple transmit and receive antennas for wireless channels: the case of digital modulation," *Global Telecommunications Conference, 2001. GLOBECOM '01.* IEEE, Volume: 6, 2001 Page(s): 3272 –3276.

[69] V. K. Goyal, "Multiple Description Coding: Compression Meets the Network," *IEEE Signal Processing Magazine*, Sept. 2001, pp 74-93.

[70] D. Comas, R. Singh, A. Ortega, and F. Marqués, "Unbalanced Multiple-Description Video Coding with Rate-Distortion Optimization," *EURASIP Journal on Applied Signal Processing" Special Issue on "Multimedia Signal Processing,"* vol.2003, No.1, Jan. 2003, pp 81-95.

[71] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, "Multiple Description Video using Rate-Distortion Splitting," *International Conference on Image Processing, 2001*, pp 978-981.

[72] F. C. Li *et al.*, "Browsing digital video," *Technical Report: MSR-TR-99-67*, Microsoft Research, Sep. 1999, ftp://ftp.research.microsoft.com/pub/tr/tr-99-67.pdf

[73] M. S. Chen, D. D. Kandlur, "Downloading and stream conversion: supporting interactive playout of videos in a client station," *Second Int. IEEE Conf. Multimedia Computing and Systems*, pp. 73-80, Washington, 1995.

[74] S. Chen, "Reverse playback of MPEG video," U.S. Patent 5,739,862.

[75] S. J. Wee and B. Vasudev, "Compressed-domain reverse play of MPEG video streams," *Proc. SPIE Conf. Multimedia Syst. and Appl.*, pp. 237-248, Nov. 1998.

[76] N. Omoigui *et al.*, "Time-compression: system concerns, usage, and benefits," *Proc. ACM SIGHI Conf.* pp. 136-143, May 1999.

[77] S. J. Wee, "reversing motion vector fields," *Proc. IEEE Int. Conf. Image Proc.*, Oct. 1998.

[78] Y. Chen, Z. Zhong, T. Lan, S. Peng, and K. Zon, "Regulated Complexity Scalable MPEG-2 Video Decoding for Media Processor," IEEE Transactions on Circuits and System for Video Technology, Vol.12, No. 8, Aug. 2002, pp678-687.

Curriculum Vitae

Jian Zhou was born in Jinan, Shandong Province, China on May 26, 1972. He received his B.S. degree and M.S. degree both from the Department of Electronic Engineering Tsinghua University, Beijing, China, in 1996 and 1999, respectively. He received his Ph.D. degree from the Department of Electrical Engineering, University of Washington, Seattle, in 2003.

From 1999 to 2003, he was a research assistant in the Department of Electrical Engineering at the University of Washington. He was an intern with Microsoft Research China in summer 2000, and with Mitsubishi Electric Research Lab in summer 2002, respectively.

His research interests include video coding, multimedia signal processing and communication.

JOURNAL PUBLICATIONS

C. Lin, J. Zhou, J. Youn, and M.T. Sun, "MPEG Video Streaming With VCR Functionality," IEEE Trans. on Circuits and System for Video Technology, Vol.20, No.3, pp415-425, Mar. 2001.

J. Zhou, H. Shao, and M.T. Sun, "Improved Transport of FGS Video for Streaming Video Applications," in preparation.

J. Zhou, H. Shao, and M.T. Sun, "FGS Enhancement Layer Truncation with Reduced Intra-Frame Quality Variation," in preparation.

CONFERENCE PUBLICATIONS

Jian Zhou, Huairong Shao, Chia Shen and Ming-Ting Sun, "FGS Enhancement Layer Truncation with Minimized Intra-Frame Quality Variation," IEEE International Conference on Multimedia and Expo, Baltimore, MD, 2003.

Jian Zhou, Huairong Shao, Chia Shen and Ming-Ting Sun, "Multi-path Transport of FGS Video," Packet Video 2003.

Chia-Wen Lin, Jian Zhou, and Ming-Ting Sun, "Minimum cost implementation of full VCR functionality in MPEG video streaming," Proc. IEEE Int. Symp. Circuits and System, Jun. 2001, Sydney, Australia.

Chia-Wen Lin, Jeongnam Youn, Jian Zhou, Ming-Ting Sun and Iraj Sodagar, "MPEG video streaming with VCR functionality," in Proc. IEEE Int. Symp. Multimedia Software Eng., pp. 146-153, Dec. 2000, Taipei, Taiwan.

PATENTS

Jian Zhou, Huairong Shao, Chia Shen, "Multi-Path Transmission of Fine-Granular Scalability Video Streams," pending.

Jian Zhou, Huairong Shao, Chia Shen, "Method for Transcoding Fine-Granular-Scalability Enhancement Layer Video to Minimized Spatial Variations," pending.