UNIVERSITY OF CALIFORNIA, SAN DIEGO

Improving end-user video quality through error concealment and packet importance modeling

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering
(Communication Theory and Systems)

by

Yueh-Lun Chang

Committee in charge:

Professor Pamela C. Cosman, Chair
Professor Truong Q. Nguyen, Co-Chair
Professor Yoav Freund
Professor William S. Hodgkiss
Professor Larry Milstein

2014

The dissertation of Yueh-Lun Chang is approved, and it is acceptable in quality and form for publication on microfilm:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2014

DEDICATION

To my family

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank everyone who helped me through my Ph.D. study journey. I am sincerely grateful to my advisors, colleagues, family, and friends for their kind support.

First, I would like to thank my advisor, Prof. Pamela Cosman, for her guidance and teaching. Her strong logical and analytical thinking has influenced me in every way. She always gave me a constructive picture when I proposed new ideas and provided thorough consideration when we discussed my project. She is an expert in writing that I learned many writing and presentation skills from her review comments. As a non native English speaker, the expression ability I gained from her is really useful for my whole life.

I would like to thank my dissertation co-chair and committee members, Prof. Truong Nguyen, Prof. Yoav Freund, Prof. William Hodgkiss, and Prof. Larry Milstein for their precious time and feedback in my PhD Qualification exam and Defense exam. The suggestions from you have strengthened the dissertation.

I would also like to thank my colleague Ting-Lan Lin, who is now assistant professor in Chung Yuan Christian University, Taiwan. When I was a junior in the lab, he gave me many good advices and guided me a lot for my very first project. I have learned many implementation skills from him, and he really helped to set a solid foundation for my later research. I also appreciate other colleagues, Yuan Zhang and Yuxia (Flora) Wang, who were visiting scholars in our lab. They shared a lot of useful knowledge with me, and we enjoyed our working and personal time together.

I would like to express my deepest gratitude to my family in Taiwan. Especially my mother, she is the one who inspires me the most to pursue this Ph.D. degree. I would have not achieved these without their encourage and support. They provided the ultimate mental shelter for me whenever I felt depressed and stressed.

Last, but not least, I would like to thank all my friends in San Diego. They are like my family in states, and we have been through many happy and difficult moments together. They have strengthened my courage to face toward my future

life journey.

Chapter 2 of this dissertation, in part, is a reprint of the material as it appears in Y.-L. Chang, T.-L. Lin, and P.C. Cosman, "Network-based IP Packet Loss Importance Model for H.264 SD Videos, IEEE Packet Video Workshop 2010. I was the primary author and the co-authors Prof. Cosman directed and supervised the research which forms the basis for Chapter 2. Prof. Lin also guided to the objective experiment in this work.

Chapter 3 of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin, Y.-L. Chang and P. Cosman, "Subjective Experiment and Modeling of Whole Frame Packet Loss Visibility for H.264", IEEE Packet Video Workshop, 2010, and in Y.-L. Chang, T.-L. Lin, and P.C. Cosman,"Network-based H.264/AVC Whole Frame Loss Visibility Model and Frame Dropping methods", IEEE Transactions on Image Processing, 2012. Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter 3. Co-author Prof. Lin also contributed to the subjective experiment in this work.

Chapter 4 is adapted from Y.-L. Chang and P.C. Cosman, "Depth-Assisted Error Concealment for I-frame loss in 2D+depth Coded Stereoscopic video", submitted to IEEE Signal Processing Letters, 2014. I was the primary author. Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter 4.

Chapter 5 is adapted from Y.-L. Chang, Y. Zhang and P.C. Cosman, "Joint Source-Channel Rate-Distortion Optimization with Motion Information Sharing for H.264/AVC Video-plus-Depth Voding" submitted to Asilomar Conference on Signals, Systems and Computers, 2014. I was the primary author. Co-author Prof. Cosman and Prof. Zhang directed and supervised the research which forms the basis for Chapter 5.

Chapter 6 of this dissertation, in part, is a reprint of the material as it appears in Y.-L. Chang, Y. Reznik, Z. Chen, P.C. Cosman, "Motion Compensated Error Concealment for HEVC Based on Block-Merging and Residual Energy," IEEE Packet Video Workshop, 2013. I was the primary author and the co-authors Prof. Cosman, Dr. Reznik and Dr. Chen directed and supervised the research

which forms the basis for Chapter 6.

VITA

2006	B. S. in Electrical Engineering, National Tsing Hua University, HsinChu, Taiwan
2010	M. S. in Electrical and Computer Engineering (Communication Theory and Systems) University of California, San Diego
2014	Ph. D. in Electrical and Computer Engineering (Communication Theory and Systems) University of California, San Diego

PUBLICATIONS

Y.-L. Chang and P.C. Cosman, "Depth-Assisted Error Concealment for I-frame Loss in 2D+depth Coded Stereoscopic video," submitted to *IEEE Signal Processing Letters*.

Y.-L. Chang, Y. Zhang and P.C. Cosman, "Joint Source-Channel Rate-Distortion Optimization with Motion Information Sharing for H.264/AVC Video-plus-Depth Coding," submitted to *Asilomar Conference on Signals, Systems and Computers*.

Y.-L. Chang, Y. Reznik, Z. Chen, and P.C. Cosman, "Motion Compensated Error Concealment for HEVC Based on Block-Merging and Residual Energy," *IEEE Packet Video Workshop*, 2013.

Y.-L. Chang, T.-L. Lin, and P.C. Cosman, "Network-based H.264/AVC Whole Frame Loss Visibility Model and Frame Dropping Methods," *IEEE Transactions on Image Processing*, 2012.

Y.-L. Chang, T.-L. Lin, and P.C. Cosman, "Network-based IP Packet Loss Importance Model for H.264 SD Videos," *IEEE Packet Video Workshop*, 2010.

T.-L. Lin, Y.-L. Chang, and P.C. Cosman, "Subjective Experiment and Modeling of Whole Frame Packet Loss Visibility for H.264," *IEEE Packet Video Workshop*, 2010.

ABSTRACT OF THE DISSERTATION

Improving end-user video quality through error concealment and packet importance modeling

by

Yueh-Lun Chang
Doctor of Philosophy in Electrical Engineering
(Communication Theory and Systems)

University of California San Diego, 2014

Professor Pamela C. Cosman, Chair
Professor Truong Q. Nguyen, Co-Chair

During video transmission, congestion and distortion in the network will cause packet loss on the video content and degrade the video quality. Traditionally the degradation is measured by mean-squared error or peak-signal-to-noise-ratio. However, these measurements do not correlate well with human perception. In this dissertation, we aim to improve the visual quality for end-users through packet importance modeling and error concealment.

The visual impact to end-users differs based on the type of the packet losses. We aim to predict how end-users respond to different losses. We start with an objective experiment in which Video Quality Metric scores are computed on fixed-sized IP packet losses for H.264/AVC SDTV video, and then construct a network-based model to predict these scores.

We would like to further understand the real visual impact on the perceptual

quality, so we conduct a human subjective experiment on whole frame losses concealed by different decoders. Whole B frame losses are introduced in H.264/AVC videos which are then decoded by two different decoders with different common concealment methods: frame copy and frame interpolation. The videos are seen by human observers who respond to each glitch they spot. It shows that even when there are more lost bits for a whole frame loss than a slice loss, the overall perceptual quality is often actually better due to the concealment that gives observers less spatial misalignment. We develop network-based models which can predict the visibility of whole frame losses. Based on the estimated visual importance, we can prioritize packets in lossy networks. The models are deployed in intermediate routers to prioritize video packets and perform intelligent frame dropping to relieve network congestion. Dropping frames based on their visual scores proves significantly superior to random dropping of B frames.

Another key solution to reduce the visual impact of video packet losses is effective error concealment methods and thus this is another focus in this dissertation. Here we work on both traditional 2D video and 3D stereo video. Among formats that provide stereo effect, 2D+depth encoding for stereoscopic video is one of the most compatible with current video content transmission systems. Traditionally the 2D and depth streams are independently coded, transmitted and concealed separately if delivered through lossy networks. We propose a new encoding scheme that offsets the I frame between the 2D and depth sequences. When a loss happens in either one, they could be concealed by the information from the other, using the strong motion correlation.

Besides providing error concealment by postprocessing at the end-user side, enhancing the error robustness of video from the encoder side is another approach. We propose an end-to-end distortion model for rate-distortion optimized coding mode selection of 2D+depth bitstreams. In our work, we first extend the encoding mode, adding an extra motion information sharing mode for the depth stream, and then improve the concealment methods. Based on these changes, we use the proposed end-to-end distortion model and derive a new Lagrange multiplier for rate-distortion optimized 2D+depth mode selection in packet-loss environments

by taking account of the network conditions, i.e. the packet loss rate.

Other than the stereo video format, a new video coding technology "High Efficiency Video Coding (HEVC)" has also been standardized in 2013. While achieving 50% bitrate reduction compared to prior standards with equal perceptual video quality, HEVC is more sensitive to packet losses since each bit contains more information. To alleviate this problem, we propose a motion-compensated error concealment method for HEVC and implement the method in reference software HM. The motion vector from the co-located block will be refined for motion compensation. Based on the reliability of these motion vectors (MVs), blocks will be merged and assigned new MVs. Our experimental result shows that not only the subjective visual quality performs well but also there is a substantial PSNR gain.

Chapter 1

Introduction

When video is transmitted through networks, it could suffer from packet losses due to various reasons, such as congestion, or bit errors. Packet losses cause perceptual degradation, but not every packet loss has equal visual impact. Some packet losses are quite visible to end-users while some are hardly noticed. Examples are given in Figures 1.1 and 1.2. In Figure 1.1(a), we show a compressed and reconstructed frame where a single horizontal row of macroblocks has been lost (the lost row is shown as a gray bar). In Figure 1.1(b) the compressed and reconstructed frame is shown where the loss has been concealed by copying the pixel values from the corresponding blocks in the previous decoded frame. The glitch in this case is visible. Figures 1.2(a) and 1.2(b) show another pair of frames with a loss and a concealed loss. In this case, no glitch is visible, because the loss occurred in a background area which was not moving.

To improve the visual quality for end-users, first we aim to measure the importance of each packet by performing objective and subjective experiments and using the data to develop models to predict the probability that each individual packet will produce an observable glitch if it is lost. The models can be deployed to prioritize video packets over the lossy network to perform intelligent dropping. Second we propose error concealment methods for several video formats to combat the situation when packet losses happen, so the video content could be recovered better for end-users. In the following, we introduce the background of quality measurement, model assessment and error concealment for video communication,

along with relevant literature.

(a) (b)

Figure 1.1: A visible case: a frame with packet loss (a) with no concealment (b) after concealment.

(a) (b)

Figure 1.2: An invisible case: a frame with packet loss (a) with no concealment (b) after concealment.

1.1 Objective video quality metrics

Subjective video quality reflects how video is perceived by a viewer and designates his or her opinion on a particular video sequence. It is the ultimate standard for video quality measurement, but the evaluation of subjective video quality is quite expensive in terms of time (preparation and running) and human resources. Several objective video quality measurements are often used instead [1].

A few common metrics are introduced in this section.

1.1.1 Mean-squared Error (MSE) and Peak-Signal-to-Noise-Ratio (PSNR)

As shown in Figure 1.3, the original image is denoted F and the reconstructed one is denoted G. For images with size M by N, the mean-squared error (MSE) is defined as follows:

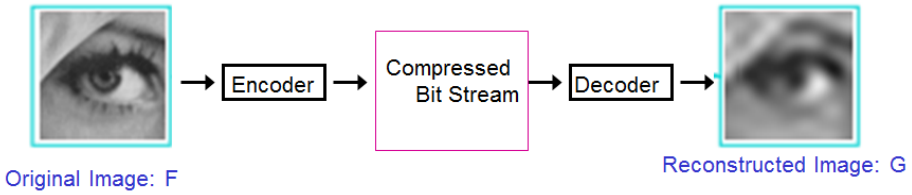


Figure 1.3: Example of the original and reconstructed images.

$$MSE = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} (F(i,j) - G(i,j))^2 \tag{1.1}$$

Peak-signal-to-noise-ratio (PSNR) is then derived by setting MSE in relation to the maximum possible value of the luminance, which is $2^8 - 1 = 255$ for 8-bit data. The PSNR value is calculated by:

$$PSNR = 10 \log_{10} \frac{(PeakValue)^2}{MSE} \tag{1.2}$$

Although several objective video quality metrics have been developed in the past few decades, PSNR continues to be a popular quality measure for pictures due to its simplicity.

1.1.2 Video Quality Metric (VQM)

Though MSE or PSNR is often used for video quality evaluation, they do not correlate well with human perception. Video Quality Metric (VQM) was developed by the Institute for Telecommunication Science to provide an objective

measurement for perceived video quality [2]. It is a computable quality metric to evaluate a processed video version in comparison with the original lossless video. It assigns a score to an entire video, or to a segment of video such as a GOP (group of pictures). VQM scores range from zero to one, where a lower score means higher quality with less degradation. It has been shown to correlate well with human perception of the video quality and has been adopted by ANSI as an objective video quality standard.

1.1.3 Structural Similarity Index (SSIM)

Instead of using traditional error summation methods, a different approach is presented in [3]. The Structural Similarity Index (SSIM) is designed by modeling any image distortion as a combination of three factors: loss of correlation, luminance distortion, and contrast distortion. The index can be calculated as

$$SSIM = \frac{(2\bar{x}\bar{y} + C_1)(2\sigma_{xy} + C_2)}{[(\bar{x})^2 + (\bar{y})^2 + C_1](\sigma_x^2 + \sigma_y^2 + C_2)} \tag{1.3}$$

where $\bar{x}, \bar{y}, \sigma_x, \sigma_y$ and σ_{xy} are the mean of x, the mean of y, the variance of x, the variance of y and the covariance of x and y. C_1 and C_2 are constants. SSIM has moderate correlation with subjective perception but it has less computational complexity than VQM.

1.2 Classification of quality assessment methods

Based on the accessibility of information about the original (reference) video, quality assessment methods can be categorized into four different types as illustrated in Figure 1.4.

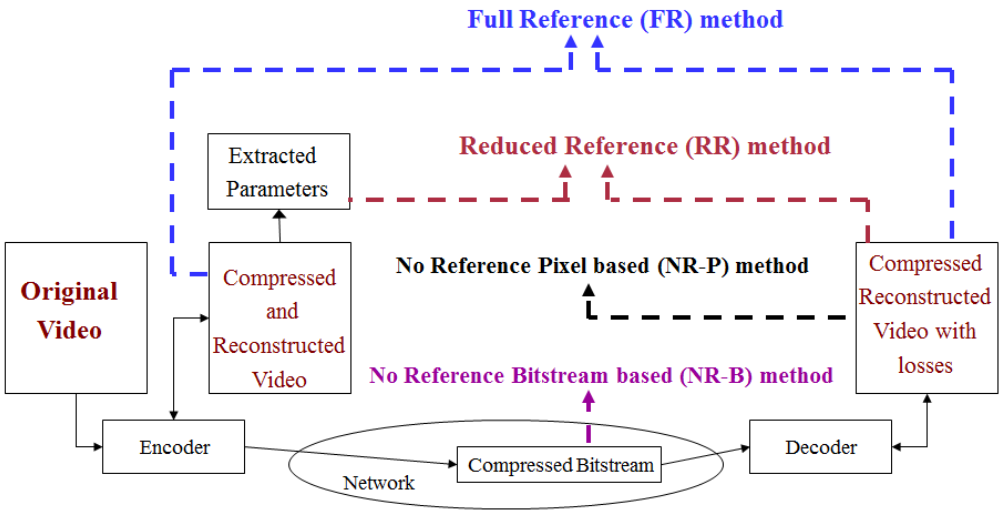


Figure 1.4: FR, NR-P and NR-B methods

- If we have access to the reconstructed video at the encoder side as well as the reconstructed video (with losses) at the decoder side, it is called a Full-Reference (FR) method. It provides the most precise measurements on the video quality difference, and the objective quality metrics introduced in Section 1.1 are all FR methods.

- If we have access to the reconstructed video (with losses) at the decoder side as well as some factors extracted from the reconstructed video at the encoder side, it is called a Reduced-Reference (RR) Method.

- If we have access only to the reconstructed video (with losses) at the decoder side, it is called a No-Reference Pixel-Based (NR-P) Method.

- If we have access only to the compressed bitstream information (with losses), then it is called a No-Reference Bitstream-Based (NR-B) Method.

According to different assessment methods, we could have two types of packet loss importance model: network-based models and encoder-based models.

A *network-based model* is built by NR methods, and it only uses information available in the bitstream or the decoded pixels without reference video. There

are many factors we could deploy in the network-based model, such as motion information, packet size, frame type, loss location, etc. However, without the ability to access the original video and to reconstruct pixel values in networks, many of the video content characteristics are not available, for example, a salient region in a frame. However, a network-based model can be deployed at different points in the network in real-time with low computational complexity. This advantage is useful to many Internet applications, such as streaming or videotelephony.

An *encoder-based model* is built by an FR or RR method, and it has access to the original video. We could have a decoder implemented at the encoder side to do all the decoding and error concealment functions which an actual decoder would do, and thus we could calculate the MSE or PSNR of pixel values between original and evaluated videos. An encoder-based model could use all of the network-based factors plus many others, such as scene cut information. With all the network and encoder factors, an encoder-based model provides the most precise prediction of packet loss importance, but it is also complicated. Some main applications for an encoder-based model are packet prioritization and unequal error protection.

In this dissertation, we will emphasize predicting packet loss importance from a network-based model and its application to intelligent packet dropping.

1.3 Error concealment methods

When video suffers from packet losses, error concealment is a postprocessing technique at the decoder to recover the damaged areas based on characteristics of video signals. To obtain a close approximation of the original pixel values and to make the reconstructed video less objectionable to end-users, several methods have been proposed [4–10]. They can be divided into two main approaches: temporal and sptial error concealment.

- *Temporal error concealment (TEC):* By utilizing blocks from other frames, TEC either reconstructs the motion vector (MV) of the lost block or searches for a block that has a good match to the sides and neighborhood of the missing block. As shown in Figure 1.5, the MVs can be simply set to zeros,

called zero motion error concealment (ZMEC) or copy, and this works well for videos with relatively small motion. The MVs can also be estimated by using the MVs of the corresponding block in the previous frame or the average of the MVs from spatially adjacent blocks, as shown in Figure 1.6.

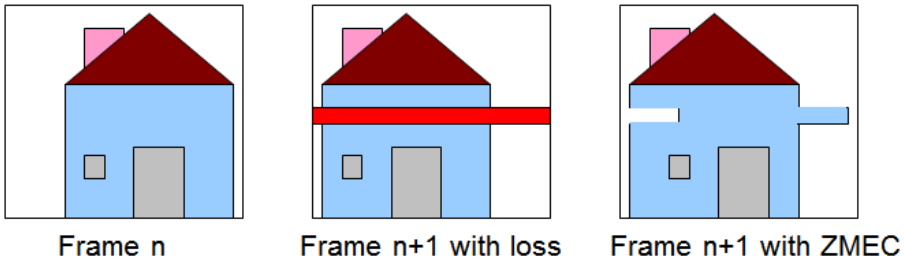


Figure 1.5: Zero-motion error concealment (ZMEC)

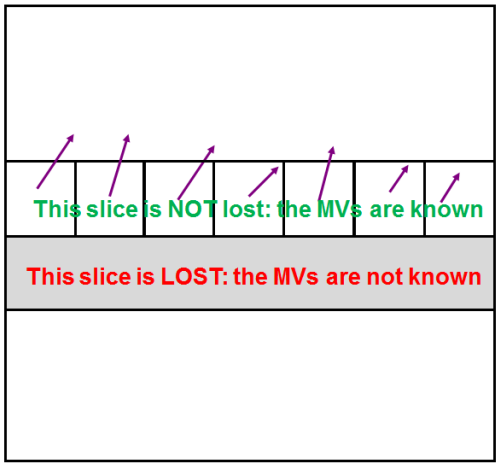


Figure 1.6: Temporal error concealment: using the MVs of the adjacent blocks.

- *Spatial error concealment (SEC):* The lost pixels can be interpolated directly in the spatial domain. For lost blocks, we can construct the damaged area with bilinear interpolation from the four nearest pixels that are not missing. Other strategies deploy directional interpolation that seeks to preserve edges. An example of SEC is shown in Figure 1.7. In general, SEC is used often on intra frames since motion estimation is not implemented in intra frames.

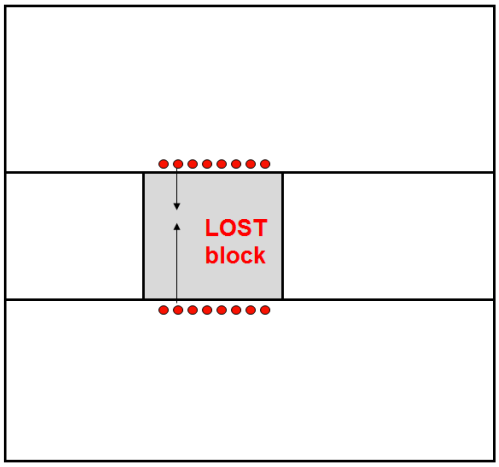


Figure 1.7: Spatial error concealment: spatial interpolation from pixels above/below the lost block .

1.4 Thesis outline

In Chapter 2, we develop a network-based packet importance model for fixed-sized IP packet loss. The objective experiment for packet importance evaluation is presented. The detailed model building strategy and the factor descriptions of the model parameters are discussed. We also analyze the effect of different IP packet losses.

In Chapter 3, we develop a network-based packet loss visibility model for whole frame loss based on a human subjective experiment. The design and setup of the experiment is introduced. We cover the analysis of data, the whole frame loss modeling process and feature selection. The model is then applied to intelligent frame dropping.

In Chapter 4, we develop a depth-assisted error concealment for whole intra frame loss in 2D+depth video. The depth map and stereo effect are introduced. We present an encoding scheme that keeps the intra frames of 2D and depth sequences offset with one another. The offset strategy provides more motion information for error concealment.

In Chapter 5, we propose a joint source-channel encoding scheme for 2D+depth

video. We impose a motion information sharing encoding scheme with an end-to-end rate-distortion model for H.264/AVC coding of 2D+depth sequences. With this encoding scheme, error concealment achieves higher PSNR without bitrate penalty.

In Chapter 6, we propose a motion-compensated error concealment method for HEVC and implement the method in reference software HM. Features of the HEVC standard are introduced. We describe the detailed concealment algorithm which combines motion vector refinement and block repartitioning.

In the Conclusions section, we summarize the contributions of this dissertation, and discuss possible future work. Partial conclusions are also given at the end of each individual chapter.

Chapter 2

Network-based slice model

In this chapter, we conduct an objective experiment in which Video Quality Metric (VQM) scores are computed on compressed video GOPs following fixed-sized IP packet loss, and then construct a network-based model to predict these VQM scores. VQM assigns a score to an entire video, or to a segment of video such as a GOP (group of pictures). The scores range from zero to one, where a lower score means higher quality with less degradation. Here we would like to develop a model to predict VQM scores using simple features that can be extracted from individual packets. The model is created for H.264 SDTV (720×480) videos using a no-reference method, meaning that we only use the information from the bitstream but have no access to the original video. The model can be computed at the packet level and requires no frame-level reconstruction.

When video is transmitted through a network, it is crucial for an intermediate router to know the visual importance of each packet to decide which ones to drop during congestion. Many of the previous research works have focused on the average quality of video subjected to an average packet loss rate. However, we would like to emphasize the influence on the video quality of an isolated or individual packet loss. Previous work [11] built a generalized packet loss visibility model using subjective tests for different encoding standards and GOP structures. The model was applied to packet prioritization for a video stream. Each packet was assigned a priority bit at the encoder so the router could perform smart dropping when the network was congested. In [12], the authors allocated more Forward

Error Correction (FEC) bits to high visibility packets to give them more protection, so as to minimize end-to-end video quality degradation due to packet losses. The models in [11] and [12] are encoder-based models, which are assessed by full-reference methods and need parameters such as MSE, type of camera motion, and information on scene cuts. These demand access to the original video at the encoder and have high computational complexity. In contrast to an encoder-based model, for a network-based model, the original video information is unavailable in the network, and the computational capability is also limited.

In addition to making a network-based model, a second goal of this chapter is to build a model for fixed-sized packets. In the network, video is typically packetized in one of two ways: it can be segmented into a variable-sized packet which contains a constant area in the frame; the other way is using fixed-sized packets which may correspond to different pixel areas but whose sizes in bits are the same, such as MPEG-2 Transport Stream packets. In previous work [13], the authors proposed a packet loss visibility model for H.264 SD and HD videos for variable-sized packets which contain one slice for each Network Abstraction Layer (NAL) unit. In this chapter, we would like to construct a network-based model for fixed-sized IP packets to predict visual importance using an objective experiment.

The chapter is organized as follows: In Section 2.1, the design of the objective experiment is described. In Section 2.2, we discuss the factors used to predict the quality scores of a loss, and the model based on these factors. Section 2.3 presents results and discussions, while Section 2.4 summarizes our conclusions.

2.1 Objective experiment on fixed-sized slice loss

In this section, we conduct an objective experiment to construct a visual importance model. Nine SD resolution H.264 videos with widely varying motion and texture characteristics are used for our experiment, and their descriptions are listed in Table 2.1. The encoder is H.264 JM9.3, and the settings can be found in Table 2.2. These settings adhere to ITU and DSL Forum Recommendations [14,15]. It is high quality compression so there are few encoding artifacts between the coded

Table 2.1: Description of video clips used for the experiment.

1	earth	nature documentary of wildlife in slow motion
2	Indianapolis	crowds moving in an arena with some car racing scenes
3	formula	racing cars on a racetrack
4	New York	introduction to a city with bird-eye and street views
5	air show	air show scene with planes flying over the sky, and some audience on the ground
6	golf	broadcast golf game
7	Hawaiian	Hawaiian tourism of various scenes in shops and streets with panning camera
8	soccer	high motion beach soccer game with crowded people in the background
9	stories	daily life such as friends talking and family reunion

videos and original ones. In these videos, each slice contains a horizontal row of Macroblocks (16×16 pixels) in a frame, and each NAL unit contains one slice. There are 300 frames in each video and the content includes various types of motion, texture characteristics and camera operations. The decoder is FFMPEG [16] due to its high efficiency and wide use in industry. For error concealment, the FFMPEG decoder begins by estimating, for each lost macroblock (MB), whether it is more likely to have been intra coded or inter coded. Based on the estimate, the algorithm uses one of two different approaches to conceal each lost MB [17].

Table 2.2: Summary of the objective experiment setup for SD videos.

Resolution	720 × 480
Bitrate	2.1 Mbps
Profile	Main profile, Level 3
Frame rate	30 fps
GOP	IBBPBBPBBPBBPBB 15/3

2.1.1 Packetization

The detailed steps to packetize H.264 SDTV videos into fixed-sized packets that can be transmitted through the network are described in this subsection.

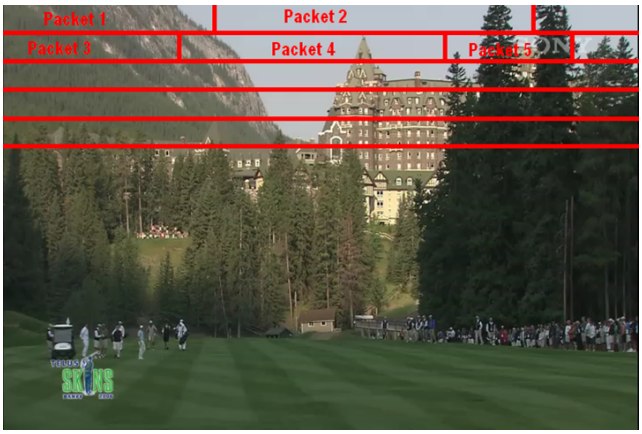


Figure 2.1: Example of transport stream packetization.

Transport Stream

The Transport Stream (TS) is defined in MPEG2-Part1 [18]. It is a digital container format that encapsulates different types of information such as video, audio or data. In [18], the authors describe how to mux several streams into a single one. The transport stream uses fixed-sized packets as its basic transfer unit. There are many advantages to using fixed-sized packets. It is convenient to detect the start and end of a frame and also easy to recover from packet loss or corruption.

A freeware tsMuxer [19] developed by the company SmartLabs is used to mux H.264 videos into regular TS packets. Each TS packet is fixed-sized with 188 bytes in length and only contains information from the same frame. An example of TS packetization is shown in Figure 2.1.

2.1.2 IP Packet

Although the transport stream specifies how to packetize multimedia information, the actual transmit unit over the network is an IP packet. Some major applications of video transmission over IP are: conversational applications such as video telephony and videoconferencing, the download of complete, pre-coded video streams, and IP-based streaming such as YouTube [20].

By the protocol specification, the size of an IP packet is variable and can

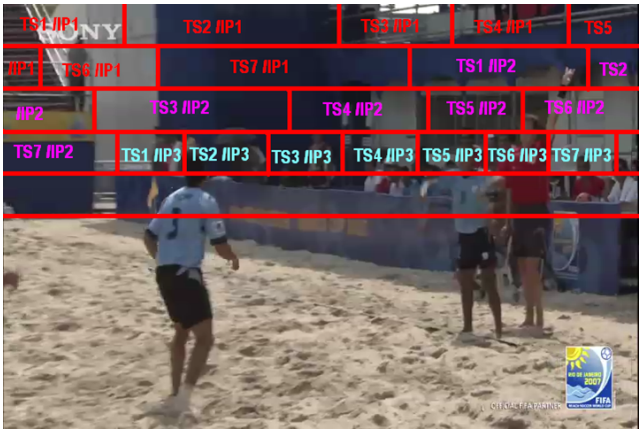


Figure 2.2: Example of IP packetization.

be up to 64 kbytes, but this size is rarely used. The reason is that a large IP packet needs fragmenting in order to pass onto the Ethernet since the payload size of the maximum transfer unit (MTU) for an Ethernet packet is 1500 bytes. To avoid splitting and recombining IP packets larger than the MTU payload size, we took the size of each IP packet to be less than 1500 bytes. Specifically, in our experiment settings, one IP packet contains seven TS packets (188×7=1316 bytes). The packet size would exceed the limitation of 1500 bytes if more than 7 TS packets were included. An example of IP packetization is shown in 2.2. Figure 2.3 shows the entire encoding and packetization process from original video to IP packets. Our goal is to construct a model to predict the VQM score associated with each IP packet, that is, the VQM score for the GOP that would result from the loss of that single IP packet.

2.1.3 Lossy Test Videos

In our experiment, we drop an IP packet from a GOP to create a lossy video and use VQM to evaluate its quality after packet loss. There are three possibilities: 1) a packet contains only one slice or a part of one slice, 2) a packet contains more than one slice, 3) a packet contains a frame header. These will cause the loss of 1) one slice, 2) several slices, and 3) an entire frame.

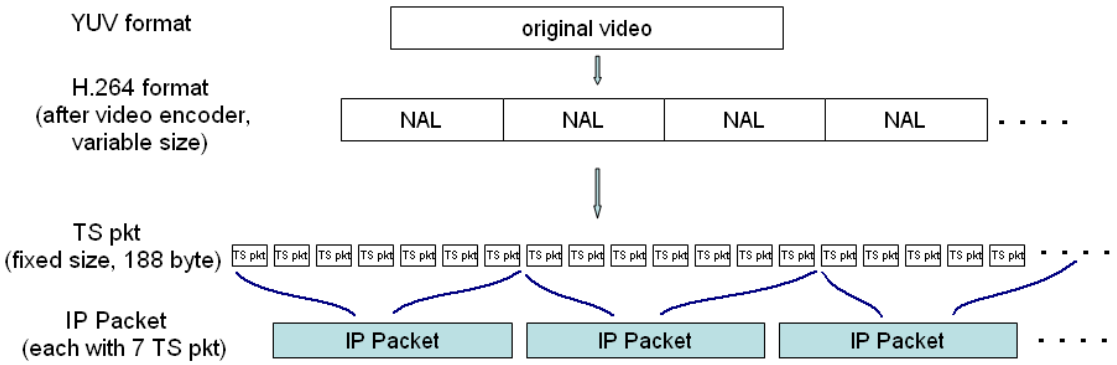


Figure 2.3: The encoding and packetization procedure from original video to IP packets.

In order to calculate a VQM score, the number of frames in the original video and in the lossy video must be the same. If a frame header is dropped, the number of frames in the lossy video cannot be kept the same. Moreover, loss of a slice header in an I frame will cause a serious degradation to the video due to the way FFMPEG decodes (the decoder does not work properly when the first slice is lost). For these reasons, the following two types of packets were considered to be the most important:

1. An IP packet with any frame header
2. An IP packet with an I slice header

Among all the IP packets from our test videos, less than 5% of them contain a frame header or I slice header, and these packets with the highest priority were not dropped in our experiment. The reason for not including these in the experiment is that our goal for predicting packet-level VQM scores is to allow a router to choose which ones to drop. For these packets of highest priority, it is already known that dropping them should be avoided if at all possible. The goal therefore is to guide the router in choosing which of the other $> 95\%$ of packets should be selected for dropping in case of congestion. After the dropping is performed for a GOP, the FFMPEG decoding and error concealment are run, and then the VQM score is calculated to obtain the objective video quality score for this GOP.

The last three frames of every GOP are excluded from being the location of the packet loss because the VQM algorithm ignores differences between the videos under comparison which occur at the end of the GOP. A total of 931 IP packet losses are divided equally and randomly among all the I frames, P frames, and B frames.

2.2 Features and model building

Our network-based model is built by using a no-reference method that only has access to the bitstream in the network while the original video is unavailable. The information we use does not require pixel data. This is desirable because the parameter extraction process can be made very efficient at a network node since it does not involve motion compensation (requiring reference frame), deblocking filter and frame reconstruction. In this section, the candidate features are described first and then the modeling approach will be explained.

2.2.1 Features

The features used to construct the model are introduced here. They can be classified into two categories: content independent and content dependent features. A buffer is used to aggregate some number of IP packets for feature extraction. Many IP packets contain no slice start code, so we have to gather information from their adjacent packets. However, there is no need for frame level reconstruction.

Content independent features only require the general information of the packet, for example, spatial and temporal location or frame type. The content independent features we considered are the following:

1. **TMDR** stands for time duration. It is the maximum number of frames that can be affected by the packet loss due to error propagation. TMDR=1 for non-reference frames. For reference frames, TMDR depends on the distance to the next I frame.

2. **DevFromCenter**=abs(Height-floor(N/2)) indicates how far the loss is from the center slice (in the vertical direction) of the frame. Height indicates the spatial location of the packet, and N is the number of slices in a frame. In our experiment, the number of slices for SDTV videos is 30.

3. **IsIFrame**, **IsPFrame** and **IsBFrame** are boolean factors which are set when the packet is in an I, P or B frame.

4. **NAL_num** is the total number of slices in the packet, and **NAL_size** is the aggregate size in bits for every slice contained in the packet. Recall that a slice is one horizontal row of macroblocks. For example, consider an IP packet which contains one partial I slice whose total size spanning across several packets is 16000 bits. For this packet, *NAL_num* is 1 and *NAL_size* is 16000. For an IP packet which contains two partial P slices whose sizes are 8144 and 11488 bits, the *NAL_num* is 2 and *NAL_size* is 8144+11488=19632.

Content dependent features require the actual content of the lost packet, such as the motion in each direction. The motion-related features take calculations over all macroblocks in the lost packet to get their mean, maximum, or variance of motion information. **MaxMotX**, **MeanMotX**, and **VarMotX** are the maximum, mean, and variance of the motion vectors in the x direction, while **MaxMotY**, **MeanMotY**, and **VarMotY** are the maximum, mean, and variance of the motion vectors in the y direction. **MaxMotA** and **MeanMotA** are the maximal and mean phase, where $MotA = \arctan(MotX/MotY)$. **MotM** is the mean magnitude of motion vectors. It equals $\sqrt{MeanMotX^2 + MeanMotY^2}$. **MeanRSENGY** is the mean residual energy after motion compensation. This is calculated from the DCT coefficients, so no inverse DCT or pixel information is needed [8]. We used the term after logarithm, and 10^{-7} is added before taking the log to avoid a log of zero problem. **MaxInterparts** is the maximal number of inter macroblock partitions in the lost packet.

To construct the model, the above features are used as well as their interaction terms, which are the products of two features.

2.2.2 Modeling Approaches

To model the VQM score, i.e., the importance of a lost packet, *Generalized Linear Models* (GLMs) provide one approach. GLMs are an extension of classical linear models [21, 22]. The packet loss importance is modeled using logistic regression, a type of GLM which is a natural model to predict the parameter p of a binomial distribution [21]. Let $y_1, y_2, ..., y_N$ be a realization of independent random variables $Y_1, Y_2, ..., Y_N$ where Y_i has binomial distribution with parameter p_i. Let $\mathbf{y}$, $\mathbf{Y}$ and $\mathbf{p}$ denote the N-dimensional vectors represented by y_i, Y_i and p_i respectively. The parameter p_i is modeled as a function of P factors. Let $\mathbf{X}$ represent a $N \times P$ matrix, where each row i contains the P factors influencing the corresponding parameter p_i. Let x_{ij} be the elements in $\mathbf{X}$. A generalized linear model can be represented as

$$g(p_i) = \alpha + \sum_{j=1}^{P} x_{ij}\beta_j \tag{2.1}$$

where $g(.)$ is called the link function, which is typically non-linear, and $\beta_1, \beta_2, ..., \beta_P$ are the coefficients of the factors. Coefficients β_j and the constant term α are usually unknown and need to be estimated from the data. Parameters x_{ij} are the features in each packet, and p_i is the expected value of the predicted term, i.e., VQM score in our experiment. For logistic regression, the link function is the logit function, which is the canonical link function for the binomial distribution. The logit function is defined as

$$g(p) = log(\frac{p}{1-p}). \tag{2.2}$$

The simplest model is a null model which has only one parameter: the constant term α. At the other extreme, the full model contains as many factors as there are data points. The goodness of fit for a GLM can be determined by its deviance, a generalization of variance. By definition, the deviance is zero for the full model, while the deviance is positive for all the other models. A smaller deviance means a better model fit. To obtain the model coefficients for the candidate factors, an iterative feature selection technique is implemented by Matlab.

To prevent overfitting, a 10-fold cross validation is applied. The data is randomly segmented into 10 groups, and we use nine out of the ten sets as the training set and the remaining as the test set. The procedure is repeated ten times, each time choosing a different set for testing.

2.3 Results and discussion

Figure 2.4 shows the histogram distribution of the actual VQM scores in our objective experiment. Higher VQM scores mean worse degradation of the video quality. In Figures 2.5 and 2.6, the plots of deviance and correlation of the actual and predicted VQM scores versus the factor numbers included are presented. While the null deviance is 71.8, the deviance of the model can be reduced to less than 45. The correlation gets higher when more factors are included. In Figure 2.6, however, there is a breakpoint when the factor number is around 10. The curve becomes nearly flat after this point, which means there is little improvement of the correlation even if more factors are added to the model.

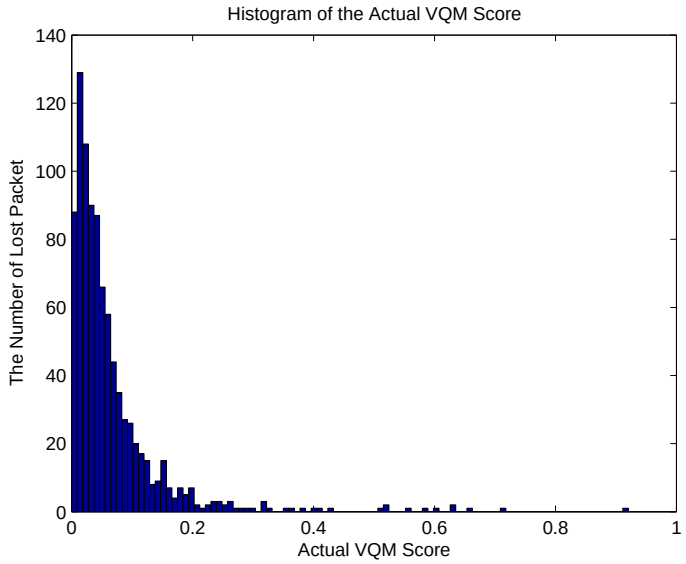


Figure 2.4: Histogram of VQM scores from the objective experiment.

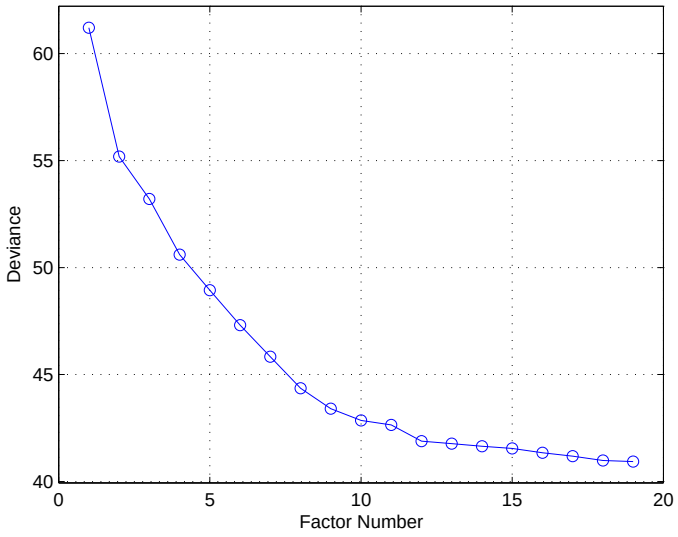


Figure 2.5: Deviance reduction as additional factors are included in the model.

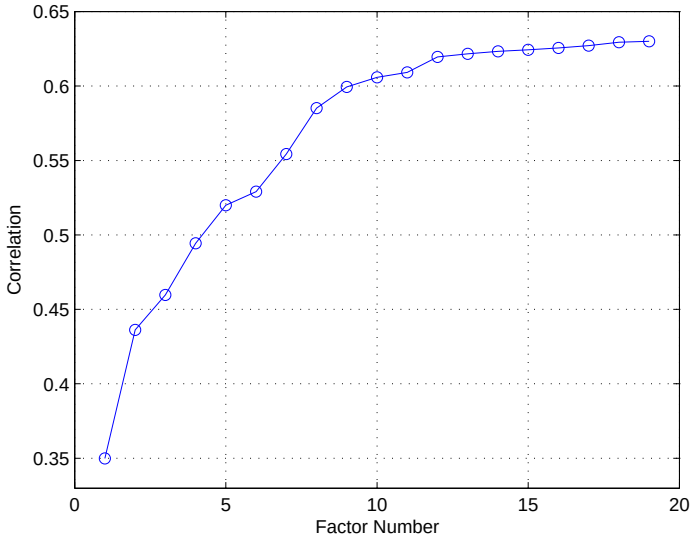


Figure 2.6: Correlation between predicted and actual VQM scores as additional factors are included in the model.

The nine most significant factors are chosen for our final model. The factors and their coefficients are listed in Table 2.3 in order of importance. The importance of a factor can be defined by the amount of deviance reduced for GLM. Each factor

Table 2.3: Table of factors in the order of importance. The × symbol means interaction.

Factor Number	Factors	Coefficients
Intercept (γ)	1	-4.1445
1	IsPFrame × log(MeanRSENGY $+10^{-7}$)	2.8457e-1
2	TMDR × NAL_size	-5.1068e-8
3	TMDR × NAL_num	3.9130e-2
4	NAL_num × IsPFrame	-2.1074e-1
5	NAL_size × MaxMotA	1.6223e-5
6	DevFromCenter × MotM	8.2868e-3
7	NAL_num × IsIFrame	1.4706
8	NAL_size × IsIFrame	-6.0151e-5
9	DevFromCenter × MaxMotA	-1.4744e-2

is the interaction of two features rather than a single term. It is sometimes hard to directly interpret the meaning of factors by the sign of the coefficients since these factors are not independent of each other. (Refer to [23], which explains why sometimes the coefficient sign is not what we expect.)

We observed the following about the effect of factors on quality:

1. The frame type of the lost packet plays a crucial role in our model, and losses in P frames were most damaging. This may seem counter-intuitive. Typically one may consider that a packet loss from an I frame would cause more degradation to the video quality, while in our model, a packet loss from a P frame actually resulted in the worst quality. This is because we packetized the video using fixed-sized packets. For SDTV videos in our experiment, an I frame generally contains 200∼400 TS packets (approximately 30∼60 IP packets), whereas a P frame contains less than 100 TS packets (approximately 15 IP packets). The detailed statistics of TS and IP packet numbers for each video are shown in Table 2.4. So one IP packet from an I frame covers on average 3.3% of the frame's area, whereas one IP packet from a P frame includes on average 13.3% of the frame's area. Sometimes the corrupted area could be as much as one-fourth or one-third of the whole frame, so the damage is worse. Examples of one IP packet loss in I frame and P frame are

given in Figure 2.7 The actual VQM scores from different frame types are shown in Figure 2.8. The histogram of VQM scores in P frames is shifted to the right compared to the histogram for I frames, and the mean VQM score resulting from a packet loss in a P frame is 0.0886, higher (worse) than that in I frames, 0.0718.

(a) (b)

Figure 2.7: Examples of one IP packet loss: (a) one IP packet loss in I frame (b) one IP packet loss in P frame.

2. Residual energy is quite important as well. Higher residual energy usually implies that the motion in the video is more complicated, or the texture is widely varied. A positive sign of the coefficient means that a packet loss with high residual energy will corrupt the video more and result in a higher VQM score.

3. Two out of the top three factors relate to TMDR. This indicates that error propagation duration is very important to determining packet loss impact on quality. Higher TMDR means that the corruption lasts longer and in general this causes worse quality with higher VQM score. Therefore TMDR should be positively correlated with VQM score. For the two factor coefficients related to TMDR, however, one has a positive sign and the other has a negative sign. These terms can be factored to single features, and the effect on TMDR is the combination of them. For example, in our model, the part

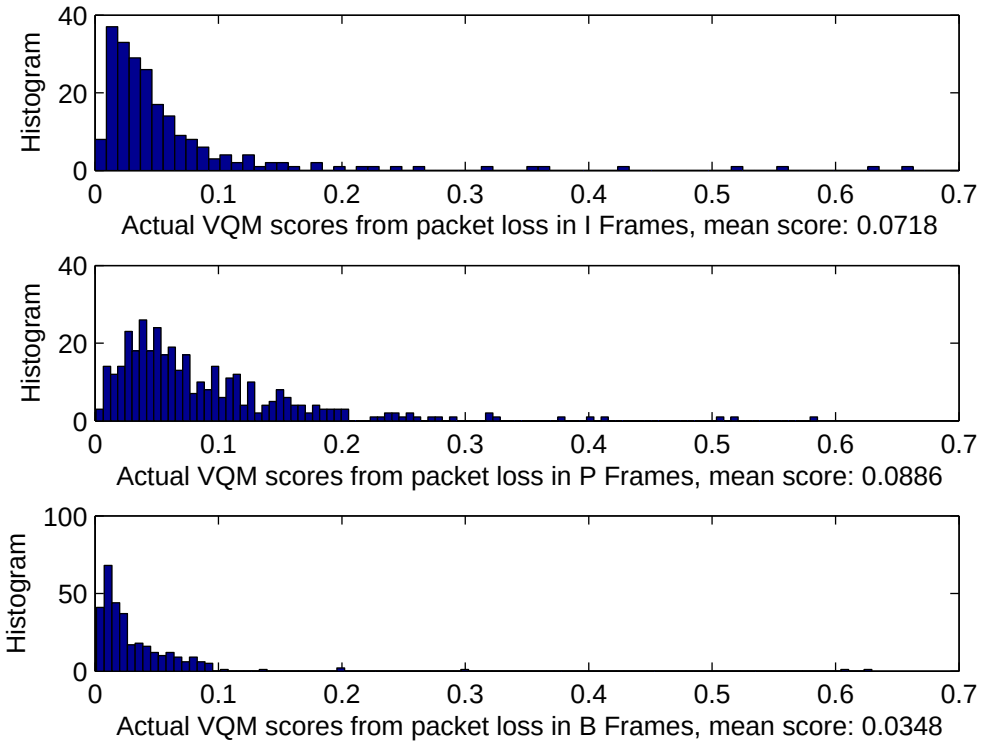


Figure 2.8: Histogram of VQM scores from different frame types.

related to TMDR is:

$$-5.1068 \times 10^{-8}(TMDR \times NAL\_size) + 3.9130 \times 10^{-2}(TMDR \times NAL\_num)$$

This can be rewritten as:

$$TMDR(-5.1068 \times 10^{-8} NAL\_size + 3.9130 \times 10^{-2} NAL\_num)$$

So the coefficient of TMDR can be considered a variable β_{TMDR}, where

$$\beta_{TMDR} = -5.1068 \times 10^{-8} NAL\_size + 3.9130 \times 10^{-2} NAL\_num$$

Considering the range of NAL_size and NAL_num, β_{TMDR} is always a positive quantity, so that TMDR has an overall positive correlation with VQM score, as expected.

4. Not only the temporal but also the spatial information is important. Six factors are associated with NAL_size or NAL_num. These terms correlate with the corrupted area within one frame, and imply that the influence of the spatial extent which could be affected by the lost packet is quite prominent.

   Since NAL_num is the total number of slices in the packet, a larger value of NAL_num means a larger contaminated area and should generally mean higher VQM score. Since we do not drop packets with an I slice header, the NAL_num of a lost packet in an I frame is always 1, while it could be any number from $1 \sim 30$ for a lost packet in a P or B frame. As we mentioned before, packet loss in a P frame usually causes the worst degradation to the quality and the highest VQM score, while the damage is less bad from a packet loss in an I frame and it is the least in a B frame. Therefore, the effect of NAL_num is separated out by the boolean features IsI/P/BFrame.

   Although the factor NAL_num appears in the model in three different interaction terms (TMDR×NAL_num, IsIFrame×NAL_num, and IsPFrame×NAL_num), the effect of NAL_num can be explained simply according to frame type.

   The coefficient of NAL_num for a loss in an I frame is a constant number $\beta_{NAL\_num\_I}$ since, for I frames, $TMDR = 15$ and $IsIFrame = 1$, so

   $$\beta_{NAL\_num\_I} = 3.913 \times 10^{-2} TMDR + 1.4706 IsIFrame = 2.0576$$

   The coefficient of NAL_num for a loss in a B frame is also a constant number $\beta_{NAL\_num\_B}$ since, for B frames, $TMDR = 1$, so

   $$\beta_{NAL\_num\_B} = 3.913 \times 10^{-2} TMDR = 0.0391$$

   Comparing the values of these two constants, we see $\beta_{NAL\_num\_I}$ is greater than $\beta_{NAL\_num\_B}$, so that the average VQM score for a loss in an I frame will be higher.

   The coefficient of NAL_num for a loss in a P frame is a variable $\beta_{NAL\_num\_P}$ depending on $TMDR$, where

   $$\beta_{NAL\_num\_P} = 3.913 \times 10^{-2} TMDR - 2.1074 \times 10^{-1} IsPFrame$$

Recall that VQM does not count quality degradation in the last three frames of a sequence, so the TMDR value for a loss in a P frame could be 6, 9 or 12. This makes $\beta_{NAL\_num\_P}$ always a positive coefficient.

In summary, the coefficients of NAL_num are positively correlated with VQM scores, which means that a larger damaging area is always worse regardless of the frame type.

5. The spatial location of the lost packet also plays a part. Analyzing the coefficient of DevFromCenter by the same method for TMDR, it generally carries a negative sign. Larger DevFromCenter means the damage is further away from the center of the video, so it is less visible.

6. MotM, the magnitude of motion, has a positive coefficient sign in the model since more movement means that a packet loss will cause a more serious degradation in quality and hence a higher VQM score.

7. Since IsIFrame, IsPFrame and IsBFrame are boolean factors and only take effect on a specific frame type, our model can be viewed in another way. Factors 1 and 4 are used in the model only for P frames. Factors 7 and 8 are used in the model only for I frames. These boolean factors (IsI/P/BFrame) construct submodels for each frame type.

2.4 Conclusion

We propose a network-based visual importance model of fixed-sized IP packets for SD H.264 videos. The proposed model allows an intermediate node in the network to efficiently estimate the visual importance of a packet by information at the packet level. Our results from the objective experiment show that, for a fixed-sized IP packet, frame type is a quite significant factor in the model. Our most novel result is finding that a fixed-sized packet loss in a P frame is on the average worse than one in an I frame. Previous studies found that I-packet losses caused the worst degradation, but that result was for packets of fixed pixel area.

Table 2.4: Table of the statistics for numbers of TS and IP packets in each video by frame type.

Video Name	Avg. number of TS / IP pkt in I Frame	Avg. number of TS / IP pkt in P Frame	Avg. number of TS / IP pkt in B Frame
Air show	319.8 / 45.6	134.2 / 19.2	70.9 / 10.1
Earth	276.8 / 39.5	89.6 / 12.8	41.7 / 6.0
Formula	233.9 / 33.4	90.7 /12.9	32.7 / 4.7
Golf	377.6 / 53.9	74.7 / 10.7	20.6 / 2.9
Hawaiian	383.7 / 54.8	78.0 / 11.1	29.8 / 4.3
Indianapolis	382.4 / 54.6	74.6 / 10.7	23.3 / 3.3
New York	369.5 / 52.8	71.9 / 10.3	21.0 / 3.0
Soccer	236.5 / 33.8	79.8 / 11.4	31.2 / 4.5
Stories	271.7 / 31.1	85.4 /12.2	26.3 / 3.8

For our packets which are of fixed size in bytes, a P-packet covers a much larger pixel area than an I-packet, and so causes more quality degradation when lost. The temporal and spatial location are also noteworthy for prediction.

Changing the fixed size of the packet or changing the resolution of the video would likely affect the model, and this would be of interest to study in the future.

Chapter 2 of this dissertation, in part, is a reprint of the material as it appears in Y.-L. Chang, T.-L. Lin, and P.C. Cosman, "Network-based IP Packet Loss Importance Model for H.264 SD Videos, IEEE Packet Video Workshop 2010. I was the primary author and the co-authors Prof. Cosman directed and supervised the research which forms the basis for Chapter 2. Prof. Lin also guided to the objective experiment in this work.

Chapter 3

Network-based whole frame model

In the previous chapter, we build a packet loss importance model based on objective experiments. However, human subjective perception is the ultimate ground truth for video quality measurement. In the research described in this chapter, we conduct a subjective experiment to gather the data for network-based whole frame models and apply the models in the networks for intelligent dropping.

While video quality monitoring in networks is an active research area, some approaches predict the video quality using objective measures such as MSE (mean-squared error) or PSNR [24–27]. However, MSE is not well correlated with human perception [28]. Therefore subjective tests collecting direct responses from subjects who watch impaired videos are necessary to understand how different packet losses are perceived by people. The work in [29, 30] focused on modeling the average quality of videos as a function of average packet loss rate. In [31], the authors developed a model utilizing mismatched blocks to predict the subjective video quality. The scene complexity and level of motion are used to predict perceptual quality in [32].

These methods give an overall quality score for the sequence, but do not tell us how to best drop packets in the router to minimize video quality degradation during network congestion. In [11], packet dropping methods based on perceptual video quality are discussed. The visual importance of each packet is evaluated

in the encoder by an *encoder-based* packet loss visibility model. All information available to the encoder can be used. Before the packet is sent to the network, a single bit of priority score is added to the header based on the estimated packet loss visibility. The router can then drop packets of lower priority during congestion. In [11], the authors showed that the dropping policy that uses visibility-based packet prioritization performs well compared to the common DropTail policy, and compared to a prioritization method based on the induced MSE if that packet is lost [33].

One limitation of [11] is that the priority score needs to be determined at the encoder and added as one bit to the packet header. In [17], the authors do not assume packets coming into the router are embedded with a visual priority bit; for each packet, the visual importance is obtained by the *network-based model* described in [13] which only requires information extractable within one packet and no reference frame information. This is desired since in a router, the incoming packets may be out of coding order or may be multiplexed with other video streams, so the router may not be able to identify which is the reference packet of the current packet. Also the authors want the complexity of the factor extraction process to be low to be used in the network. Therefore the authors do not consider factors such as initial mean square error or scene cut detection that require pixel domain reconstruction by full decoding as used in [11].

Also in [17], the authors devise a packet dropping method for widely varying packet loss rates including high rates. The packet loss visibility modeling was designed for packets that contain individual slices (defined to be one horizontal row of macroblocks) of a frame. For these slice losses, after error concealment, spatial misalignment relative to the intact portion of the frame stands out. Spatial misalignment artifacts can be more distracting than temporal frame freeze [34]. Therefore in [17], the algorithm drops the least visible *frames*, incurring fewer blocky artifacts compared to dropping on a *slice* basis. The authors showed that the frame-level temporal interpolation artifact is better than the slice-level spatial misalignment artifact using the VQM [35]. VQM is a full-reference metric that considers jerky motion, blocking, and blurring [36], and has been shown to correlate

well with human perception [37].

Nevertheless, which whole frame to be dropped in [17] was estimated by the network-based visibility model for single-slice packets described in [13]. That is, the visibility score for the frame was taken to be the sum of the visibility scores for the slices which compose the frame. And those visibility scores for slices came from a model designed using a human observer experiment on slice loss data, which do not directly reflect the frame loss visibility. This chapter aims at obtaining and exploiting more meaningful scores for frame losses. We conduct a subjective experiment on whole frame loss, and build a direct model for whole frame loss. Two common concealment methods are used for whole frame losses: frame copy and temporal frame interpolation. We analyze the experimental data, and model the whole frame packet loss visibility based on information associated with the lost frames. We use the model to intelligently drop frames, and compare performance with [17] and [38].

Perceptual quality of frame losses is also discussed in the literature; [39] concludes that viewers preferred a single but long freeze event to frequent short freezes. In [40], different whole frame loss types were studied as a function of frame loss burst length and distribution. The authors conclude that the visibility of frame dropping is dependent on content, loss duration and motion. Later, in [41], they built an assessment model for subjective video quality as a function of frame loss burst length and distribution. However, the quantities are computed in the pixel domain and require the original video, and the model aims to evaluate the quality of an entire lossy video, and does not indicate the visual importance of a specific frame.

This chapter is structured as follows: in Section 3.1, the setup of the subjective experiment is introduced. Section 3.2 covers the analysis of data, and Section 3.3 introduces the whole frame loss modeling process and feature selection. Section 3.4 proposes frame dropping algorithms using the whole frame loss visibility model and the frame size, and gives the performance of various methods. Section 3.5 concludes the chapter.

3.1 Subjective experiment on whole frame losses

In this section, we introduce the subjective experiment setup, including the encoding configuration, decoder concealment and experimental design. The video encoder is H.264/AVC JM 9.3. Encoder settings (Table 3.1) adhere to ITU and DSL Forum Recommendations [14, 15]. Each Network Abstraction Layer (NAL) packet contains a horizontal row of Macroblocks (16×16 pixels) in a frame. Our tested resolution is SDTV, so we have 30 packets per frame. Nine videos we used in Chapter 2 are concatenated into a 20-minute sequence.

The decoders we considered are the JM 9.3 standard decoder [42] which produces frame copy artifacts, and FFMPEG [16] which conceals whole frame losses using temporal frame interpolation. For the JM decoder, the lost frame is concealed by copying the pixels from the previous reference frame. For the FFMPEG decoder, a lost P frame is concealed by copying the pixels from the previous reference frame, and a lost B frame is concealed by temporal interpolation between the frame pixels of the previous and the future reference frames. These two decoders are widely used in academia and industry.

In this experiment, we concentrate on B frames. We introduce whole frame loss events once every 4 seconds to allow observers enough time to respond to each individual loss event. There are two types of whole frame loss events: single whole frame loss and dual whole frame loss; every loss event occurs in the first 3 seconds of each 4-second interval. Among these intervals, we uniformly inject single or dual whole frame losses in a GOP (in the dual cases, the *distance between the two lost*

Table 3.1: Summary of the subjective experiment setup. H is the height of the video.

	SDTV
Resolution	720×480
Bitrate	2.1 Mbps
H.264 Profile	Main profile Level 3
Viewing Distance	6H
Frame rate	30 fps
GOP	IBBPBBPBBPBBPBB 15/3

frames in one GOP could range from 1 to 13).

We create six different realizations of whole frame loss events of the 20-minute video, producing 900 distinct single whole frame loss events and 900 dual whole frame loss events. All the six lossy videos are decoded by FFMPEG and JM decoders. A subject watches two different loss realizations of whole frame loss events from the same decoder, so a session involves 40 minutes of actual watching time per subject. The experiment takes one hour or less, including an introductory session and a break. When viewers see a glitch, they respond to it by pressing the space bar. If the response is within 2 seconds of the loss, the loss event is regarded as visible. Each of the 40-minute lossy videos is watched by 10 people.

The ground truth loss visibility score for a specific loss event is calculated as the number of people who see the loss artifact divided by 10. Since there are six different realizations of the lossy videos and each is watched by 10 subjects, we have a total of 60 people participating in the experiments, where 30 people watch JM-decoded videos and 30 people watch FFMPEG-decoded videos. For each type of loss event, 1800 ground truth visibility scores are obtained (900 for the JM decoder and 900 for the FFMPEG decoder).

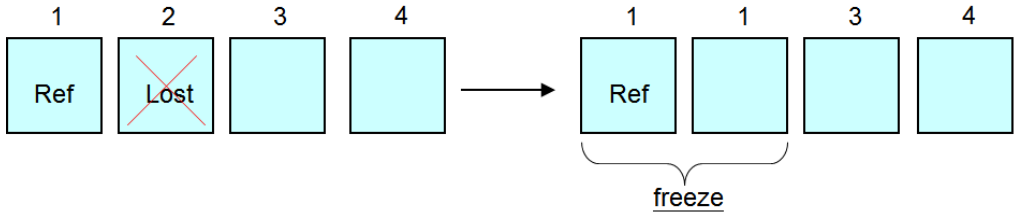
3.2 Data analysis

In this section, we analyze the two types of whole frame loss events: single frame losses and dual frame losses. We examine the artifacts caused by the different concealment methods of the JM and FFMPEG decoders, and then compare the performance of the decoders.

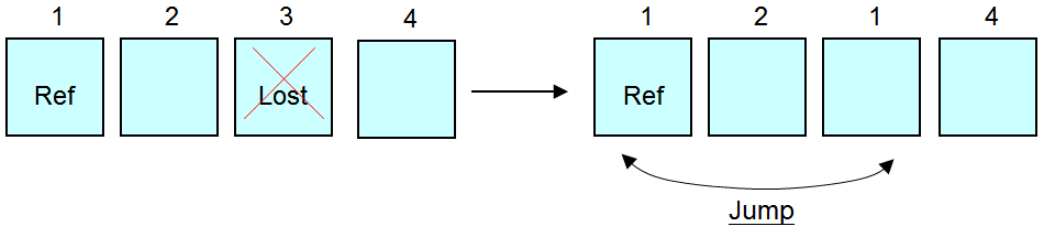
3.2.1 Concealment methods of the decoders

JM uses frame copy and FFMPEG uses temporal interpolation for whole frame loss concealment. For all B frames, JM conceals them by copying the previous intact reference frame, causing two types of temporal concealment artifacts: freeze and jump. For example, in Figure 3.1(a), Frame 2 is lost and Frame 1 is the reference frame for Frame 2. Frame 2, if lost, is concealed by copying Frame

1; the visual artifact is a short freeze since Frame 1 is displayed twice, in two consecutive frame time slots. In contrast, in Figure 3.1(b), if Frame 3 is lost, it is also concealed by copying Frame 1. The displayed frames are 1, 2, 1, 4 rather than 1, 2, 3, 4, which causes jerkiness or jumping visually. The FFMPEG decoder conceals B frames by temporal interpolation most of the time, except for B frames after an IDR frame which are concealed by copying the IDR frame. For temporal interpolation, ghosting artifacts may appear when there is enough motion. The above three types of artifacts are called *"freeze"*, *"jump"* and *"interpolation"* effects. A visual example is demonstrated in Figure 3.2. Frame 35 of video sequence "Stefan" is lost and concealed by JM with frame copy in Figure 3.2(a) and by FFMPEG with temporal frame interpolation in Figure 3.2(b).



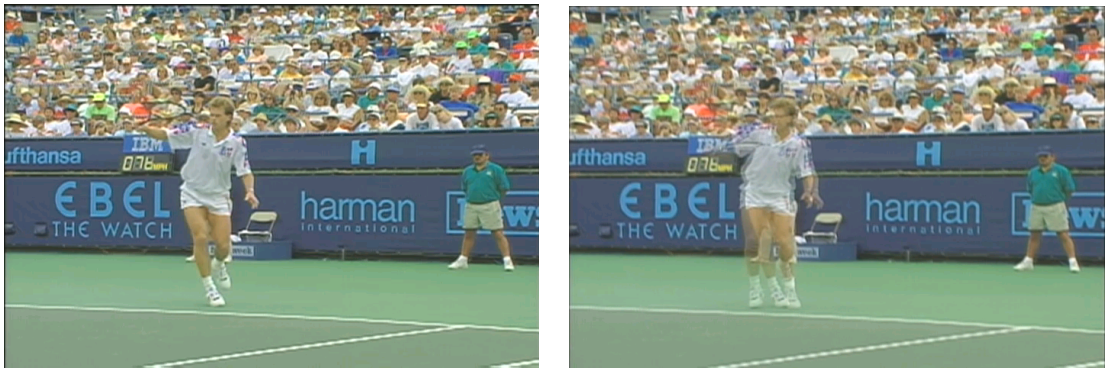
(a) freeze effect



(b) jump effect

Figure 3.1: Different visual effects by frame copy concealment: (a) freeze effect and (b) jump effect

Table 3.2 shows the mean visibility for the three types of effects, calculated from the single whole frame loss events. The freeze effect has the lowest mean visibility of 0.07, the jump effect has the highest of 0.29, and the visibility of



(a) (b)

Figure 3.2: Frame 35 of video sequence "Stefan" is lost and concealed by the JM decoder with frame copy in (a) and by the FFMPEG decoder with temporal frame interpolation in (b)

Table 3.2: Three types of artificial effects and their corresponding mean visibility, calculated from the single whole frame loss events

Effects	Mean visibility
Freeze	0.07
Jump	0.29
Interpolation	0.19

interpolation is intermediate at 0.19.

Table 3.3 summarizes all the possible artifacts of dual whole frame loss concealment for each decoder, and the corresponding mean visibility for each is also shown. Figure 3.3 shows the visibility for different concealment artifacts. What is plotted in each case is the mean visibility together with the 95% confidence interval. The cross markers are for single frame losses, while the circle markers are for JM dual frame losses and the triangle markers are for FFMPEG dual frame losses. The 95% confidence intervals for the single frame loss concealments are non-overlapping, meaning that the three effects (freeze, jump and interpolation) have significantly different visibility. On the other hand, some of the 95% confidence intervals for the dual frame loss concealments are overlapping because the artifacts are the combination of two effects. The artifacts with jump effect have relatively higher mean visibility while the artifact with mere freeze effect has the lowest

Table 3.3: Possible artifacts for concealed dual whole frame losses and the corresponding mean visibility for both JM and FFMPEG decoders.

Decoders	Possible artifacts	Mean Visibilty
JM	a freeze effect of three frames	0.22
	a jump effect and then a freeze effect	0.28
	a freeze effect and then a jump effect	0.25
	two freeze effects	0.08
	two jump effects	0.38
FFMPEG	a freeze effect of three frames	0.26
	an interpolation effect and then a freeze effect	0.21
	a freeze effect and then an interpolation effect	0.24
	two interpolation effects	0.26
	a jump effect and then an interpolation effect	0.37

visibility. The loss events with interpolation effect give an intermediate result. About 30% of events are not seen by any observers, and on average 2.4 out of 10 observers see a dual frame loss event.

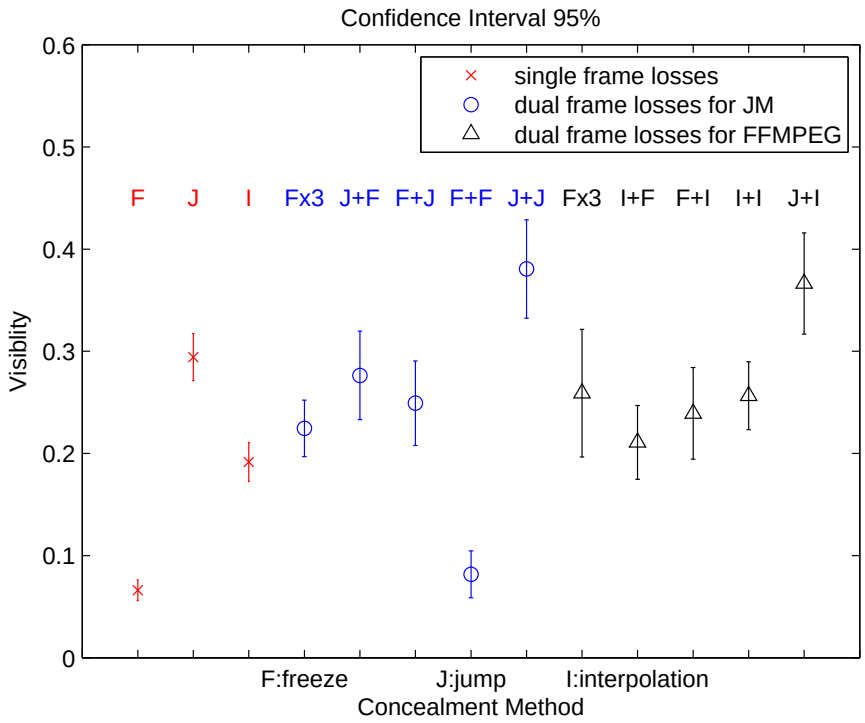


Figure 3.3: Whole frame loss visibility showing means and 95% confidence intervals, for different concealment artifacts.

We also look into the dual frame loss visibility versus frame distance, as plotted in Figure 3.4. In our experiment, the frame distance for the two nearby whole frame losses in one GOP ranges from 1 to 13. The mean visibility is periodically higher for frame distances equal to 4, 7, 10 and 13. In the dual frame loss events, for a certain frame distance there are several possible frame loss combinations, which result in different artifacts since the concealments are not the same. For instance, when frame distance equals 1, the visual artifacts are either two freeze effects or two interpolation effects. The mean visibility of each frame distance is a weighted average of the visibility for the various dual frame loss concealments which can occur at that spacing. Statistically, when frame distance equals 4, 7, 10 and 13, their frame loss combinations result in a larger percentage of jump effect compared to other frame distance cases, and it makes these four frame distance cases have higher mean visibility.

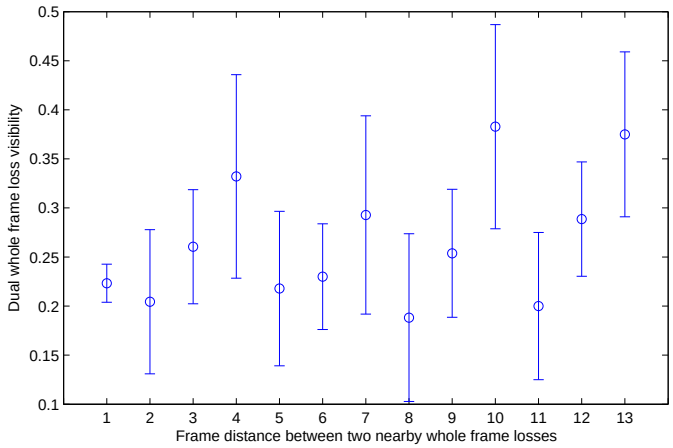


Figure 3.4: Dual whole frame loss visibility showing means and 95% confidence intervals, for every frame distance

Another way to analyze the visibility is to group events into adjacent dual frame losses and separate dual frame losses. The two lost frames are adjacent if the frame distance equals 1, while they are separate if the frame distance is greater than 1. Figure 3.5 shows the dual frame visibility for the adjacent and separate cases. It is apparent that adjacent dual frame losses have lower visibility than separate dual frame losses since the adjacent cases only lead to the two less visible effects (freeze and interpolation) while the separate cases can lead to the jump effect.

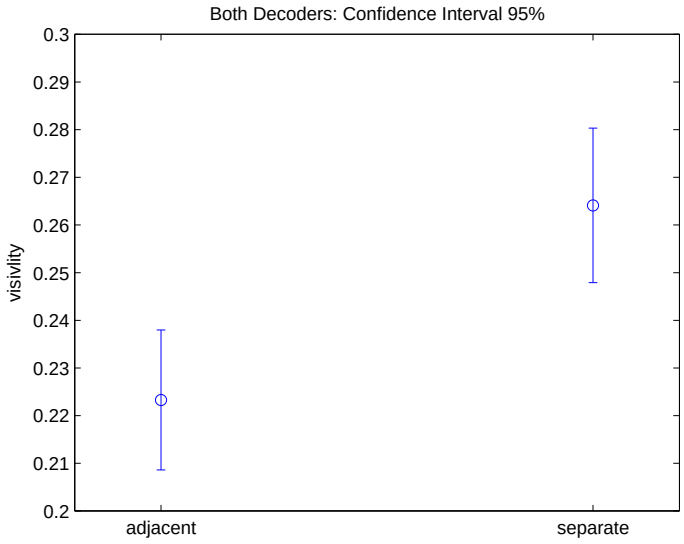
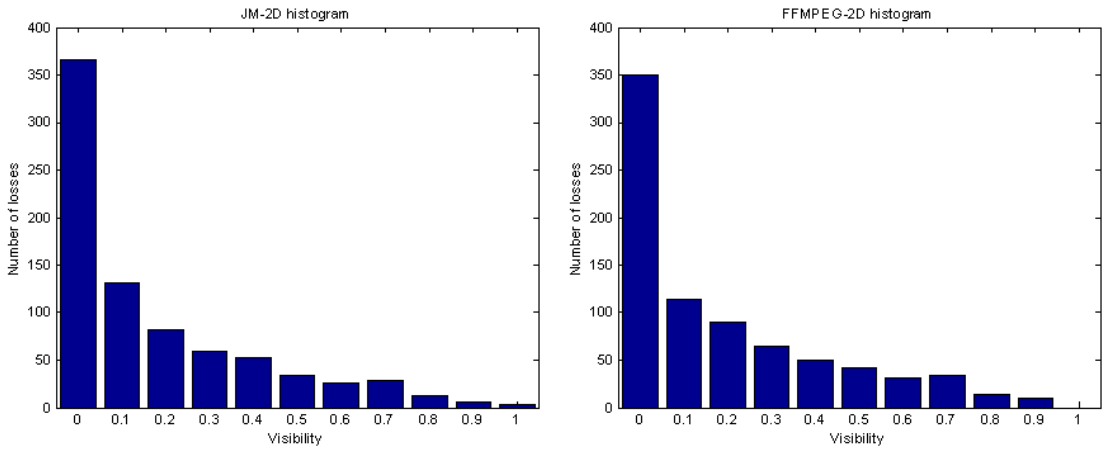


Figure 3.5: Dual whole frame loss visibility showing means and 95% confidence intervals, for the adjacent and separate cases

3.2.2 Comparison of the decoders

In this section, we compare the performance of the JM and FFMPEG decoders only for single frame losses since we would like to build models that predict visibility for an isolated frame loss. Figure 3.6 shows the histograms of the single whole frame loss visibility of the JM and FFMPEG decoders. For the JM decoder, 40.8% of the losses are not observed by any subjects (visibility is zero), and 62.4% of losses are seen by 2 or fewer out of 10 people (i.e., have visibility less than or equal to 0.2). For the FFMPEG decoder, 38.9% of the losses are not observed by any subjects, and 58.3% have visibility less than or equal to 0.2. One implication is that if we can identify these frames that are less visible to viewers when lost, in the case of network congestion, we can choose to drop unimportant frames to relieve network congestion, and not many end users will observe the losses.

In the design of our experiment, because there is a loss event in every 4 second interval, it could be a concern that viewers would begin to anticipate the next loss event. However, we do not believe that viewers noticed the loss pattern because there was such a high percentage of loss events which were invisible, so



(a) (b)

Figure 3.6: Histogram of single whole frame loss visibility by (a) JM decoder, (b) FFMPEG decoder.

viewers were not perceiving losses in each time slot.

Figure 3.7 is the 3-D histogram of the single whole frame loss visibility with respect to the JM and FFMPEG decoders. This figure shows that the invisible whole frame losses decoded by JM usually are also invisible by FFMPEG and vice versa. The JM decoder has a better score than FFMPEG on 33.2% of cases, and FFMPEG has a better score 29.6% of the time. The remaining 37.2% of the whole frame losses are observed by exactly the same number of observers for JM and FFMPEG. Among the tie cases, 79% represent losses with zero visibility for both decoders. The average whole frame loss visibility over all the data is 0.1716 for JM and 0.1879 for FFMPEG, indicating that on average, whole frame losses concealed by JM are slightly less visible than by FFMPEG.

For a significance test between the visibility scores of FFMPEG and JM, we can not perform a hypothesis test that assumes the data to be normal (e.g., t test) since from Figure 3.6, their distribution is far from normal. Therefore we resort to nonparametric hypothesis testing. The Wilcoxon Signed Rank Test (paired comparison) [43] compares paired data x and y in a two-sided test where

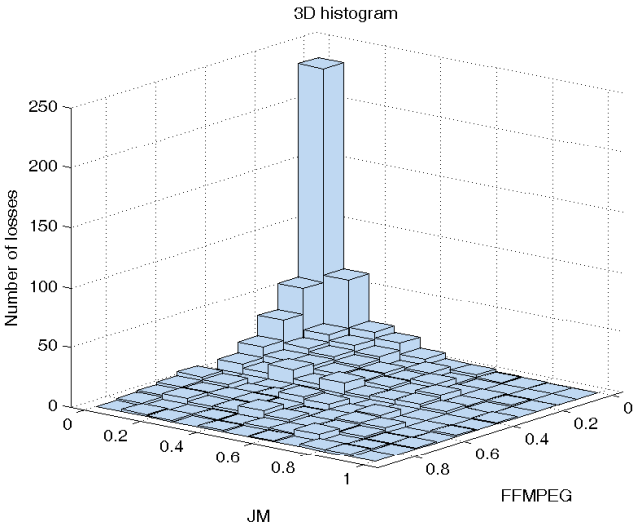


Figure 3.7: 3-D Histogram of single whole frame loss visibility by JM decoder and FFMPEG decoder

the null hypothesis H_0 is that the median of $x - y$ is zero, against the alternative that the distribution does not have zero median. Let x_i and y_i be the visibility for FFMPEG and JM in the ith comparison set. Define $w = \sum_{i=1}^{n} r_i z_i$ where r_i is the rank of $|x_i - y_i|$ among all $|x_j - y_j|$, and $z_i = 1$ if $x_i - y_i > 0$ and $z_i = 0$ otherwise. Here $n = 900$, the number of losses. The statistic for the test,

$$Z = \frac{w - [n(n+1)]/4}{\sqrt{[n(n+1)(2n+1)]/24}}, \tag{3.1}$$

distributes approximately as Normal(0,1) when $n > 12$. The p-value is 0.176 ($> 5\%$), meaning that we cannot reject the null hypothesis at the 95% confidence level that the visibility scores of FFMPEG minus JM come from a distribution of zero median. From the previous section, we know that the freeze and jump effects by JM cause the best and the worst visibility while the interpolation by FFMPEG gives an intermediate result. This evens out the overall performance of the two decoders so there is no significant difference between the visibility of JM and FFMPEG. This motivates us to develop one model to predict the whole frame packet loss visibility for both decoders. We discuss this in the next section.

3.3 Whole frame packet loss visibility model

In this section, we construct a prediction model for whole frame loss visibility using the data from the single whole frame loss events. To predict the loss visibility, we consider network-extractable factors associated with a particular frame computed from a bitstream. The process of model building and feature selection will be discussed.

3.3.1 Factors extractable from bitstream for predicting frame loss visibility

From a frame, we want to obtain factors that can be extracted without the need for other frames. Therefore, we do not consider initial MSE and other metrics involving operations related to pixel domain reconstruction (as pixel reconstruction would require access to the reference frame). By this, the frame loss visibility can be determined even in the case that we do not have access to other frames.

Several factors are shown to be important to the prediction of slice loss visibility in our prior study [11, 13]. For each MB in a frame, there are seven features that we extract or compute from the bitstream. These are **RSENGY** (the residual energy after motion compensation, obtained from the DCT coefficients), **QP**, **Interparts** (the number of partitions of the MB), and four motion-related parameters: motion in x and y directions, magnitude of motion (**motM**) and angle of motion (**motA**). For each of these seven quantities, we include the mean, maximum, and variance of the values (computed over all MBs in the frame) as predictive features in our model. To compute motA, we only consider MBs with non-zero motion, for which the phase is well defined. We also include the mean, maximum, and variance of the slice sizes as predictive factors. For residual energy, as in [11], we found that this factor after logarithm was more correlated with frame loss visibility (where we add 10^{-7} before taking the log to avoid a log of zero problem). Therefore we use this transformation.

In addition, MB modes might affect the frame loss visibility, thus we include the number of MBs that are coded as INTRA (**NumIntraMB**), INTER

(**NumInterMB**), DIRECT (**NumDirectMB**) and SKIP (**NumSkipMB**) as model factors. To include in a simple way the effects of concealment, we defined boolean factors **IsFreezeByJM**, **IsJumpByJM**, **IsInterpolation**, **IsFreeze-ByFFMPEG** and **IsJumpByFFMPEG** which are set when the certain effect is possibly present for a frame. These five concealment-related factors could be obtained by knowing the temporal location of a frame.

The motion information mentioned above is estimated by the network node where reference frames are assumed to be unavailable; in some cases, the "true" values for those quantities require the reference frames. For example, the "direct" mode of coding a macroblock assumes that an object is moving with constant speed, so the motion vector for the current MB is copied either from the spatial neighborhood or from the previous co-located MB. Within a frame, we do not have any information on the previous co-located macroblock. We instead copy the motion vector from a spatial neighbor. This way, the model is fully self-contained at the frame level, and can be implemented at a network node.

3.3.2 Modeling Process

As before, we choose a GLM with the logit function as link function to predict the packet loss visibility, since it can predict a probability parameter in a binomial distribution. We assume each viewer's response is an independent observation of the average viewer (for whom we are developing the model). Therefore, each viewer response can be considered i.i.d. with probability p for seeing a particular packet loss.

To prevent overfitting, a 4-fold cross validation is applied. The data is randomly segmented into 4 groups, and we use three out of the four sets as the training set and the remaining as the test set. The procedure is repeated four times, each time choosing a different set for testing. We perform the feature selection process on the responses collected from the subjective experiment and the factor set described in Section 3.3.1, plus interaction terms between any two factors in the set by multiplication between two factors.

If we have information about the user and know the exact decoder to be

deployed, we could build models based on different decoders: $JM_$Model and $FFMPEG_$Model. Figures 3.8 (a) and (b) show the plots of deviance versus the number of factors included in the model. The concealment-related factors greatly improve the deviance. Because most of the losses in the $FFMPEG_$Model are concealed by temporal interpolation with interpolation effect, the concealment-related factors benefit the $JM_$Model more since they correctly depict the visual effects of freeze or jump, which are both caused by frame copy but with very different influence on the visibility.

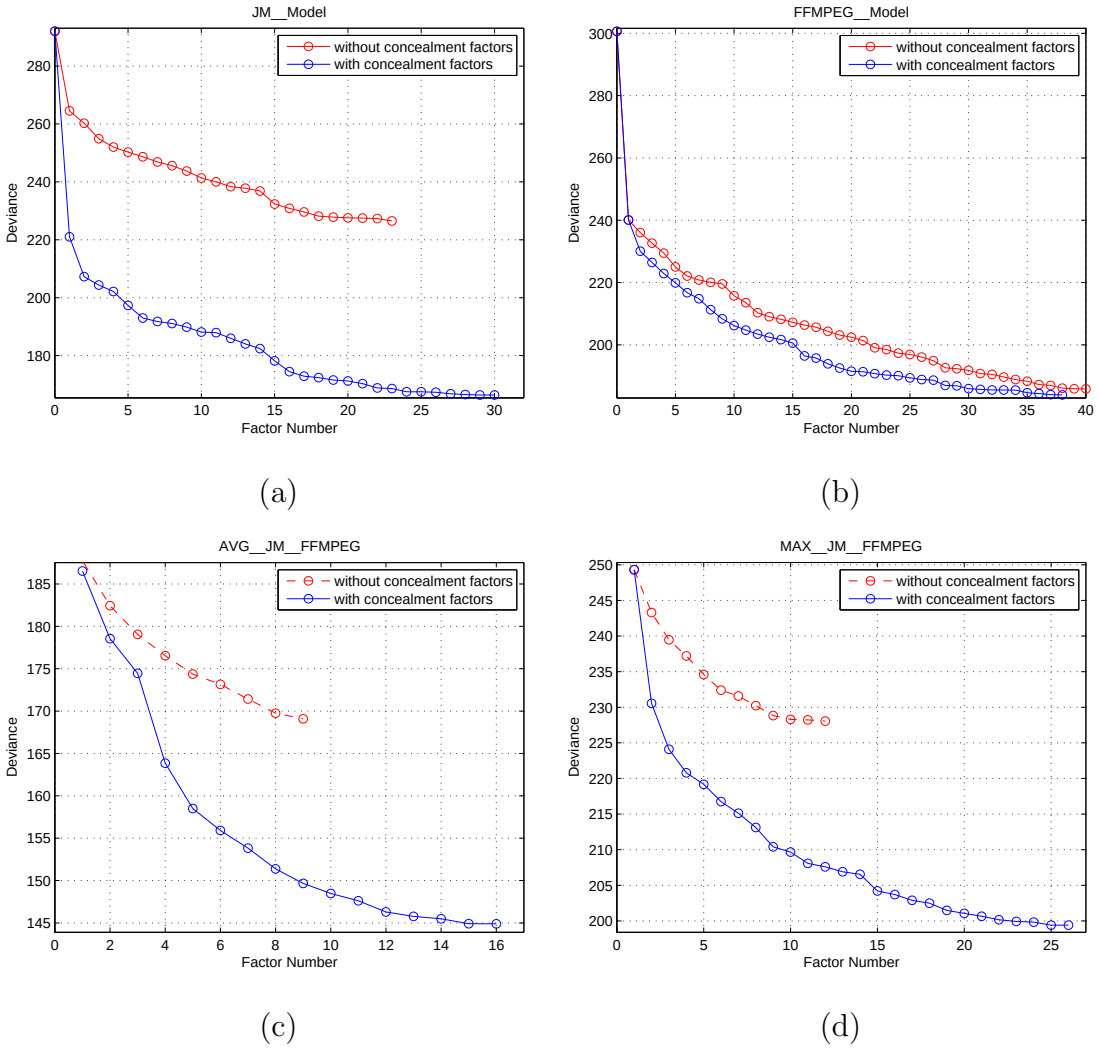


Figure 3.8: Deviance reduction as additional factors are included in the (a) $JM_$Model (b) $FFMPEG_$Model (c) Avg_JM_FFMPEG (d) Max_JM_FFMPEG model

In case one does not know at an intermediate router which decoder will be used ultimately at the receiver side, it is desirable to develop one model to predict the whole frame packet loss visibility for both decoders. The data is combined in two ways: taking the average of the JM and FFMPEG visibility scores associated with the same whole frame loss, and taking the maximum of the JM and FFMPEG visibility scores; the latter aims to predict the worst case visibility. The factors in order of importance and the corresponding coefficients of the final models of Avg_JM_FFMPEG and Max_JM_FFMPEG are listed in Tables 3.4 and 3.5 respectively. Their plots of deviance versus the number of factors included are shown in Figure 3.8(c) and (d).

Table 3.4: Table of factors in the order of importance for Avg_JM_FFMPEG model.

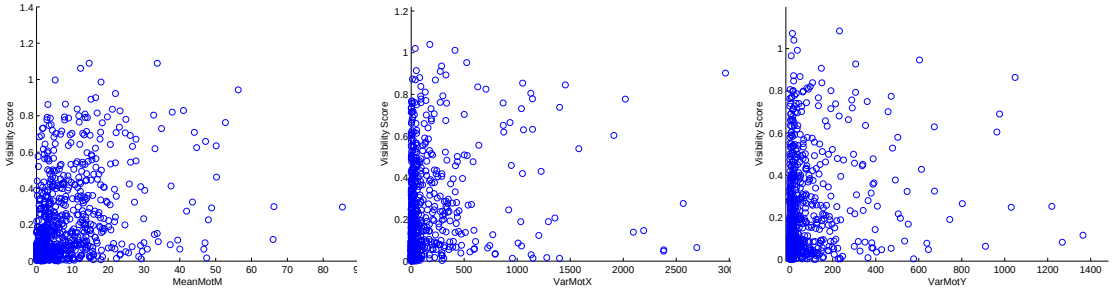
Order	Factors	Coefficients
α	1	-3.8051
1	IsJumpByJM × MeanMotM	-2.7522e-2
2	log(VarRSENGY $+10^{-7}$)	1.6276e-1
3	IsJumpByJM × MaxMotA	4.4779e-1
4	MeanMotM	1.0879e-1
5	VarMotY	-2.9205e-3
6	MeanSliceSize × IsJumpByFFMPEG	7.6570e-05
7	VarMotX	-2.1337e-3
8	VarMotM	2.2820e-3
9	IsInterpolation × MaxMotY	-8.3836e-3
10	IsFreezeByJM × MeanMotY	-2.5011e-2

The first seven important factors are almost the same for both models, but with a slightly different order. More than 70% of factors in the model involve motion vector computations. This indicates the amount of motion in the lost frame dominates the visual performance. Figure 3.9 shows the scatter plots of visibility score versus three of the top important factors: MeanMotM, VarMotX, and VarMotY. Since the visibility scores take on only 11 discrete values (0, 0.1, 0.2, ... 1) which cause the dots to overlap in the scatter plot, we add random values between 0 and 0.095 to each visibility score for plotting. So the points with visibility score of 0 are shown with y values randomly between 0 and 0.095, those with values

Table 3.5: Table of factors in the order of importance for Max_JM_FFMPEG model.

Order	Factors	Coefficients
α	1	-3.7488
1	MeanMotM	9.4095e-2
2	IsJumpByJM $\times$ MaxMotA	5.6668e-1
3	VarMotY	-1.5806e-3
4	MeanSliceSize $\times$ IsJumpByFFMPEG	9.6291e-05
5	IsInterpolation $\times$ MeanMotA	-9.1844e-2
6	log(VarRSENGY $+10^{-7}$)	7.9889e-2
7	VarMotX	-7.1111e-4
8	MaxMotM	9.4269e-3
9	MaxMotY	-2.7974e-3
10	IsJumpByFFMPEG $\times$ MeanMotM	-3.7718e-2

of 0.1 are shown with y values in the range of 0.1 to 0.195, etc. This makes it easier to see the distinct dots. The trend in the plots shows that the visibility tends to be larger when the three factors have higher value; the dots tend to be more tightly clustered at the low visibility side when the factor values are small. As in the separate model, the concealment-related factors are important. Without these concealment-related factors, the best deviance for the Avg_JM_FFMPEG and Max_JM_FFMPEG models are only 171 and 229, considerably higher than when concealment-related factors are included. The 9 video clips used in the subjective experiment included both high and low motion; we found that the model accuracy was slightly higher for slow motion clips than for high motion clips.



(a) (b) (c)

Figure 3.9: Scatter plots of visibility score versus three of the top important factors: (a)MeanMotM, (b)VarMotX, and (c)VarMotY.

3.4 Whole frame dropping

In this section, we discuss an application of the whole frame visibility model. We consider a situation in a network where the incoming video rate at a router is higher than the outgoing rate. The router should perform video data dropping to maintain the video quality as much as possible. If the router can accurately measure the visual importance of each piece of data, it can decide what to discard.

In our experiment, bit reduction rate (BRR) is defined as the percentage of bits that need to be dropped of the buffered data to alleviate the congestion. We use the whole frame loss visibility models from the previous section to determine the visual importance of the frames and design a dropping protocol. To achieve better video quality under the constraint of a target dropping rate, the size of the frame should be considered along with estimated visual scores.

3.4.1 Dropping algorithms under comparison

We use the proposed models in Tables 3.4 and 3.5 that directly predict the whole frame visibility to perform the frame importance estimation. The model in Table 3.4 is used to predict the frame importance, and drop frames until the target BRR is achieved. This method is denoted **FrameMean**. When the model in Table 3.5 is used, we denote it **FrameMax**.

If there are two frames of the same size, to minimize the visual impact of frame dropping, it is intuitive to drop the one with lower visual score. However, if there are two frames of different sizes but with the same visual scores, it is better to drop the frame with larger size. To include the size consideration, we drop frames with least ratio of visual score to size. For the methods of FrameMean and FrameMax, these versions are denoted **FrameMeanBit** and **FrameMaxBit**. The experimental results show that this concept improves the video quality.

As a baseline for comparison, [38] discusses a dropping method that is implemented in a video-aware digital subscriber line access multiplexer (DSLAM). It inspects the nal_ref_idc (NRI) bit in every NAL unit header. Packets which do not serve as reference pictures can be dropped during network congestion. That corresponds to B frames in our case. We simulate this method by randomly dropping B frames until the BRR is achieved. We denote this method by **RandomBFrame** and define its performance by the results averaged over 50 random realizations. A variation that considers size drops B frames in descending order of size. This dropping method is denoted **LargestBFrame**.

3.4.2 Experimental results

In this section, we compare the six methods for different videos and different levels of BRR. The lossy bitstreams which result from each dropping method and network condition are decoded by the FFMPEG and JM decoders.

The video encoder is H.264/AVC JM9.3. The resolution is SDTV. The tested videos are encoded at 2.5Mbps, 30 fps using Main profile Level 3. The GOP structure is IBBP (18 frames). We perform each dropping algorithm in a GOP, and the BRR is the percentage of bits to be dropped for this GOP. After the dropping policy is performed for a GOP, the FFMPEG and JM decoding and their corresponding error concealment are run, and then the VQM score is calculated to obtain the video quality score for this lossy GOP.

Eight videos are tested in the simulation; they contain a wide variety of scenes with different types of camera motion, object motion and spatial texture. *Golf* has slow movement, and *Soccer* has fast motion; these two videos are among

the sequences used in the subjective experiment in Section 3.1. Other clips are *News*, *Mother Daughter*, *Opening*, which have low motion, and *Stefan*, *Table Tennis*, *Whale* with high motion; these standard test videos were not used in the subjective experiments.

The simulated BRRs are 0.5%, 5%, 7.5%, 10%, 15% and 20%. Note that BRR can be very different from packet loss rate (PLR). For example, 20% BRR can result in 50% PLR if the dropping algorithm drops B packets, which have much smaller sizes than I or P packets on average. Therefore, BRR ranging from 0.5% to 20% considers a very wide range of packet dropping levels. The BRR of 0.5% causes only one frame loss most of the time. In this condition, RandomBFrame by averaging over 50 random realizations could perform better than LargestBFrame because deterministically selecting the largest B frame to drop will generally exceed the 0.5% dropping target and not correspond to the lowest visbility. Based on this, when BRR equals 0.5%, we do not use LargestBFrame and other visbility-per-bit methods.

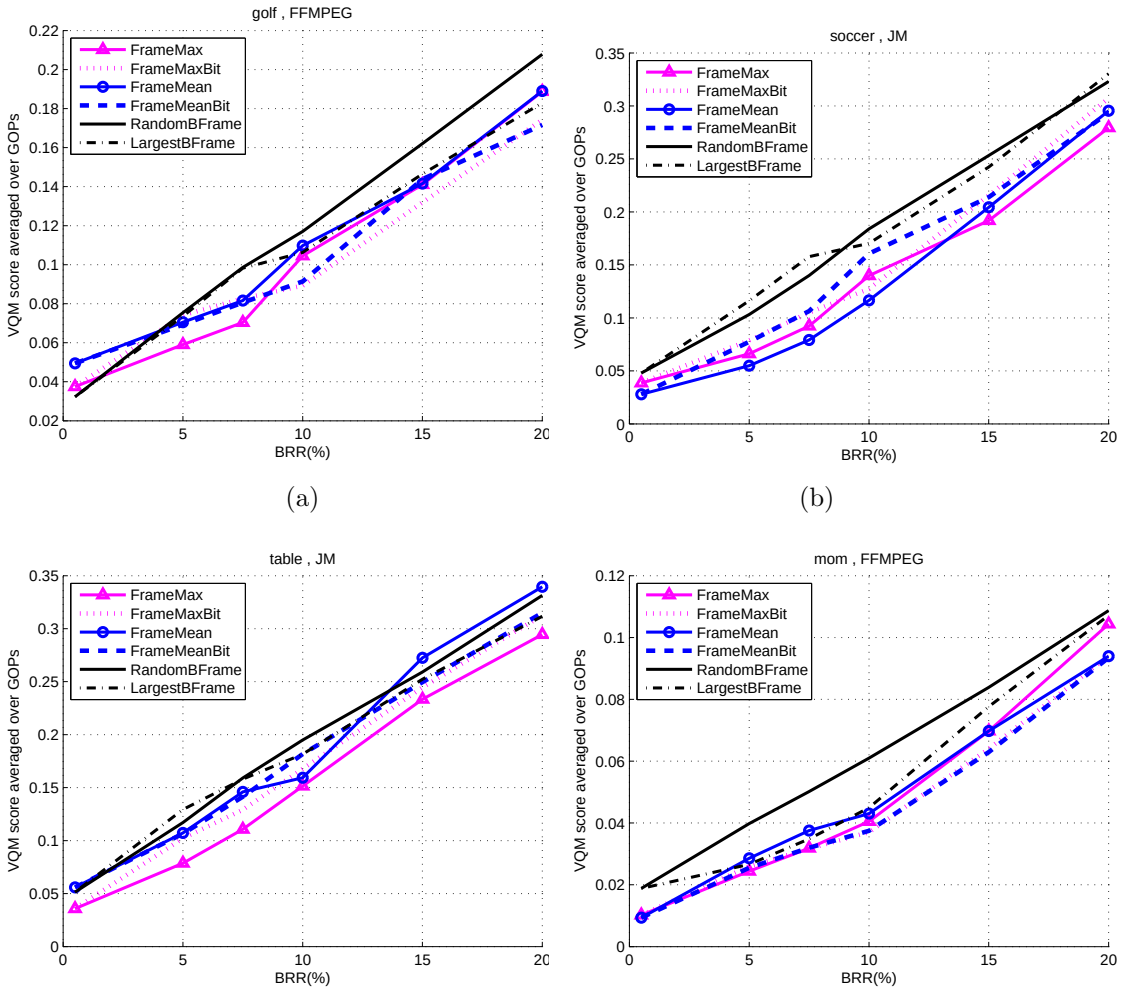
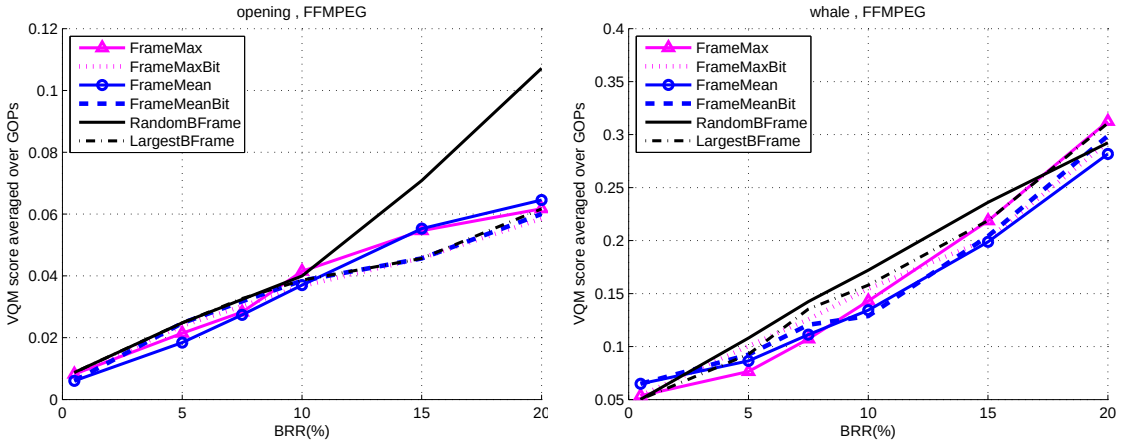


Figure 3.10: Average VQM score over GOPs vs. BRR for the six packet dropping policies for (a) FFMPEG for *Golf*, (b) JM for *Soccer*, (c) JM for *Table tennis*, (d) FFMPEG for *Mother Daughter*, (e) FFMPEG for *Opening* and (f) FFMPEG for *Whale*. Lower VQM scores correspond to higher quality.



(e) (f)

Figure 3.10: Average VQM score over GOPs vs. BRR for the six packet dropping policies for (a) FFMPEG for *Golf*, (b) JM for *Soccer*, (c) JM for *Table tennis*, (d) FFMPEG for *Mother Daughter*, (e) FFMPEG for *Opening* and (f) FFMPEG for *Whale*. Lower VQM scores correspond to higher quality. Continued.

Figure 3.10(a) shows VQM score averaged over GOPs versus BRR for the six dropping methods for the video *Golf*, where the lossy bitstream is decoded by FFMPEG. We see that as the BRR goes up, the video quality deteriorates (the VQM scores go up).

The non-visibility based methods RandomBFrame and LargestBFrame are compared first. LargestBFrame beats RandomBFrame most of the time, so we obtain a good VQM improvement by knowing the size of each frame. Especially for network nodes with low computation ability, this primitive method could provide some benefit.

We then compare the visibility based methods. FrameMean and FrameMax perform better than our prior method in [17], which means the models directly built from whole frame losses provide a better prediction of frame importance than estimating frame importance by summing the visibility of slices in a frame using the slice loss visibility model. In addition, the visibility-per-bit methods provide further improvement. FrameMeanBit and FrameMaxBit are better than FrameMean and FrameMax respectively. These trends can be observed in Figure 3.10(a).

For all other videos shown from Figure 3.10(b) to Figure 3.10(f), there are similar trends. For the comparison between the visibility and the visibility-per-bit

methods, it is not consistent that one of them is superior; however, in more than half of the cases, the visibility-per-bit method outperforms the visibility method.

Comparing the low motion clips (*Golf*, *News*, *Mother Daughter*, *Opening*) and high motion ones (*Soccer*, *Stefan*, *Table Tennis*, *Whale*), the slow movement clips have lower VQM scores than the fast movement clips for a given BRR. In the simulation, the highest VQM scores for the fast clips are more than 0.3, while the highest scores for the slow ones are less than 0.25. This indicates that the losses are more concealable for *Golf*, *News*, *Mother Daughter*, and *Opening*. Comparing the best dropping approach with the worst one, the fast motion videos have larger gains. In Figures 3.10(b), (c) and (f), the improvement for high motion videos increases more, up to 0.05∼0.08 VQM score whereas the slower videos have less than 0.04 VQM score gain as in Figures 3.10(a), (d) and (e). It is important to note that even the worst dropping method, RandomBFrame, requires some packet inspection and decoding of a slice header within a packet. If a router does not do any inspection, dropping effects would be worse.

3.5 Conclusion

In this chapter, we describe the results of a subjective test on whole frame loss and concealment, the construction of models predicting the loss visibility, and a packet dropping experiment based on these models. The contributions of this research can be summarized as follows:

1. When isolated B frames were lost and concealed by either the JM standard decoder or the FFMPEG decoder, about 40% of such losses were not seen by any of the ten observers, and about 60% of such losses were seen by two or fewer out of ten observers. This suggests that whole frame loss of isolated B frames is highly concealable.

2. Although the JM and FFMPEG decoders had very similar overall performance, this result hides the fact that, depending on frame position, JM concealment produces freeze or jump artifacts, whereas FFMPEG concealment produces mostly interpolation artifact (and only rarely a freeze or jump

artifact if a B frame after an IDR is lost). And these concealment approaches do not have similar performance, as freeze is the least noticeable, and jump is the most visible.

3. When two B frames are lost within the same GOP, about 30% of such events are not seen by any observers. On average 2.4 out of 10 observers see a dual frame loss event. The least visible type of dual frame loss event consists of two isolated freeze artifacts. So if a router needs to drop two frames within a GOP, the best choice would be to have two separate pairs of B frames in a GOP each suffer the loss of the first B frame in the pair. This leads to the least visible type of loss for the JM decoder, and among the least visible for FFMPEG.

4. Visibility models which are specific for the JM and FFMPEG decoders are more successful at predicting the frame loss visibility than are models which attempt to predict the average or the worst case of the two decoders. Nonetheless, a model designed to predict the average visibility score can provide improved frame dropping decisions compared to random B frame dropping, and compared to slice-based visibility dropping decisions from [17].

5. In the condition where an intermediate router is congested and is forced to drop frames (needing to achieve some target bit reduction rate), if for complexity reasons one does not wish to drop frames using the visibility model, there are still ways to improve over random B frame dropping. One way is to drop the largest B frames until the target is met; this offers improvement especially for larger bit reduction rates because one achieves the target with a smaller number of total frames dropped. A second simple way to improve over random B frame dropping is to avoid dropping the second B frame in any pair of two consecutive B frames (this avoids the jump concealment artifact).

Chapter 3 of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin, Y.-L. Chang and P. Cosman, "Subjective Experiment and Modeling of Whole Frame Packet Loss Visibility for H.264", IEEE Packet Video Workshop, 2010, and in Y.-L. Chang, T.-L. Lin, and P.C. Cosman,"Network-based

H.264/AVC Whole Frame Loss Visibility Model and Frame Dropping methods", IEEE Transactions on Image Processing, 2012. Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter 3. Co-author Prof. Lin also contributed to the subjective experiment in this work.

Chapter 4

Depth-assisted error concealment for whole intra frame loss in 3D video

So far we have focussed on analysis and application of packet loss importance modeling in the network. When packet losses happen, error concealment at the decoder side is one solution to reduce video quality degradation for end-users. In this chapter, we propose a depth-assisted error concealment for whole intra frame loss in 2D+depth video.

The chapter is organized as follows: In section 4.1, an overview of 2D+depth video format is introduced. In Section 4.2, the proposed algorithm is described. Section 4.3 presents the experimental results and discussion, while Section 4.4 summarizes our conclusions.

4.1 Overview of 2D+depth video format

Stereo effect is generated by creating the illusion of depth in an image by means of stereopsis for binocular vision. Most stereoscopic methods present two offset images separately to the left and right eyes of the viewer. These 2D images are then combined in the brain to give the perception of 3D depth. Two of the major 3D video formats are: multi-view video coding (MVC) and 2D+depth [44].

Multiple view compression has been standardized as the MVC extension of H.264/AVC [45, 46]. An MVC coder consists of N parallelized single-view coders. Each of them uses temporal prediction structures, where motion compensated prediction (MCP) is deployed. The inter-view dependency of these picture are also utilized for disparity compensated prediction (DCP). The simplest MVC will have two views, i.e., left and right viewS, and the example of motion and disparity compensated prediction for MVC are shown in Figure 4.1. MVC was adopted to the 3D Blu Ray specification for coding 2-view stereo in 2009, but one of the main restriction for MVC is the linear dependency of the coded data rate from the number of cameras [47]. This makes MVC not that applicable when it has a higher number of views.

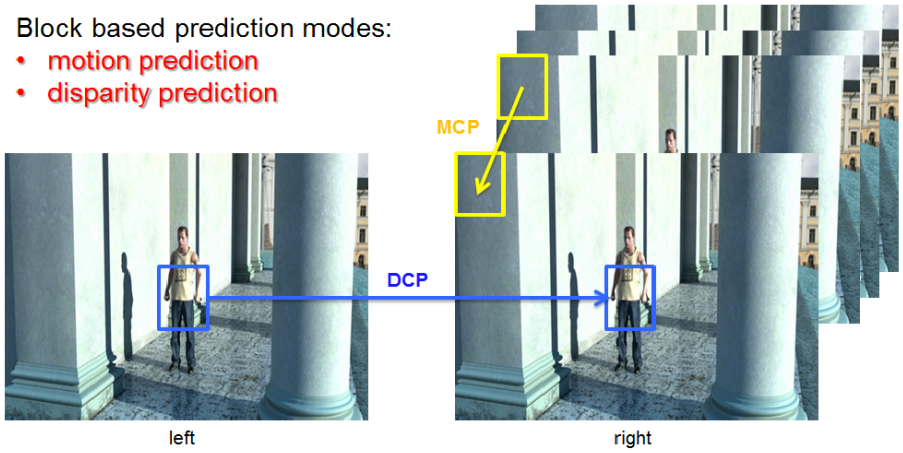


Figure 4.1: Example of motion and disparity prediction for MVC.

Another popular 3D video format is multi-view video plus depth format (MVD) [48]. By adding scene geometry, i.e., depth maps, MVD not only has limited bitrate increase, but also allows rendering arbitrary numbers of additional views via view synthesis. Depth data can be obtained in different ways. One can estimate the depth value based on the acquired pictures or use special sensor to record low-resolution depth maps, i.e., time-of-flight cameras. Synthetic sequences made with 3D models can be re-rendered in stereoscopic 3D by adding a second virtual camera, for example, computer generated scene content and animated films.

A 2-view example, 2D+depth, is shown in Figure 4.2. In this format, only one monoscopic video stream and an associated per pixel depth sequence need to be encoded. The 2D sequence provides the surface, the color, the structure of the scene, while the depth sequence represents the distance between the optical center of the camera and a point in the visual scene. New views can be synthesized using various depth image-based rendering (DIBR) approaches [49] at the decoder side.

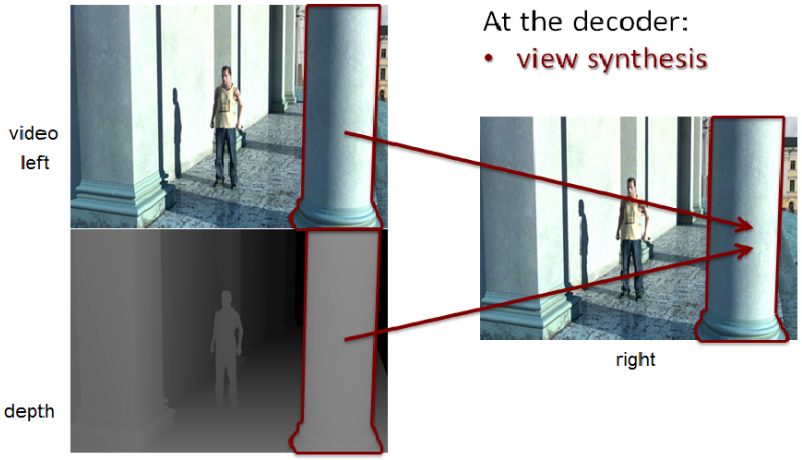


Figure 4.2: Example of 2D+depth format for MVD.

A depth map can be thought of as a gray image, with characteristics very different from normal 2D images. Depth images contain little texture and contain predominantly flat patches with sharp edges marking boundaries between objects at different depths. In [50], the 2D video and the depth map were shown to be spatially correlated, and the MVs in the two sequences are highly correlated. However, it is the 2D video that dominates the view synthesis quality of the 2D+depth format [51]. An overview of 2D+depth video transmission system is shown in 4.3

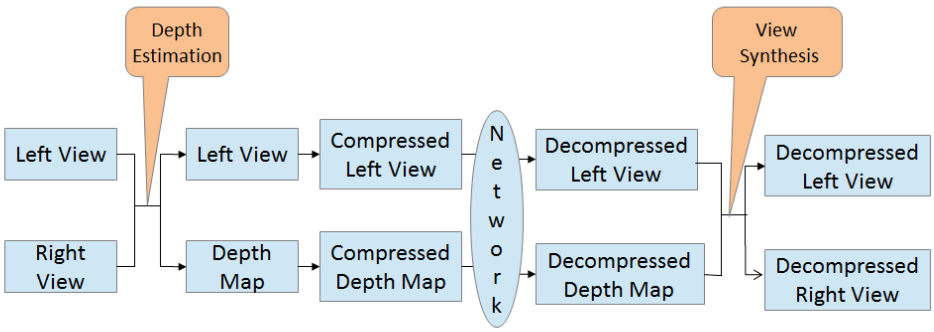


Figure 4.3: Overview of 2D+depth video transmission system.

Whole frame loss is a troublesome problem in transmitting sequences over error-prone networks. Especially when a loss happens to an intra frame in the 2D sequence, it causes more damage than a loss in the depth sequence. Conventional error concealment methods for intra frame loss are spatial interpolation [52] or pixel copy from the co-located region in the previous frame. A better solution is to use the temporal correlation of the sequence, such as decoder motion vector estimation (DMVE) [53] and data hiding [54]. These methods have high computational complexity and are not suitable for intra frame loss in 2D+depth sequences because they do not exploit the motion similarity. To ameliorate this problem, previous work [55, 56] present a depth-offset encoding scheme to make 2D intra frames correspond to inter-coded ones in the depth sequence. The MVs from the inter frames could help conceal the intra frame slice loss. This work focused on slice loss where the lossy region is only part of a frame and it relied on the boundary match algorithm (BMA) [57]. It will not be applicable when an entire intra frame is lost, since no remaining pixels could serve as the criterion for boundary distortion measurement.

4.2 Proposed depth-offset with motion compensated encoding

As a stereoscopic video coding format useful for 3D displays, 2D+depth coding is not yet standardized. Usually the 2D and depth sequences are coded independently with aligned GOP structure, ignoring the fact that they are spatially associated and the motion vectors are highly correlated. MV correlation mostly benefits inter-coded frames rather than intra-coded ones. As an intra frame loss generally causes more serious error propagation compared to an inter frame loss, error concealment that could properly utilize the motion similarity is desirable for 2D+depth sequences.

Previous work [55, 56] imposed a depth-offset encoding scheme. The first GOP of the depth sequence is shortened while the rest of the GOPs remain the same length. The temporal offset will have intra frames of the 2D and depth sequences not aligned with each other; every intra frame in the 2D sequence corresponds to an inter frame in the depth sequence. Thus the authors could retrieve the MVs from the depth sequence if a 2D intra frame slice is lost and deploy the information to assist error concealment, using either boundary matching or a hybrid procedure.

In this section, this depth-offset encoding scheme is refined with motion information sharing between the intra frame in the 2D sequence and its corresponding inter frame in the depth sequence. As shown in Figure 4.4, to attain an accurate motion description of the scene, an additional motion estimation is performed on the 2D intra frame. Instead of coding this motion information in the 2D stream, the frame is still solely intra-coded, but the MVs are delivered to the depth sequence. The corresponding inter frame is then motion compensated based on these shared MVs.

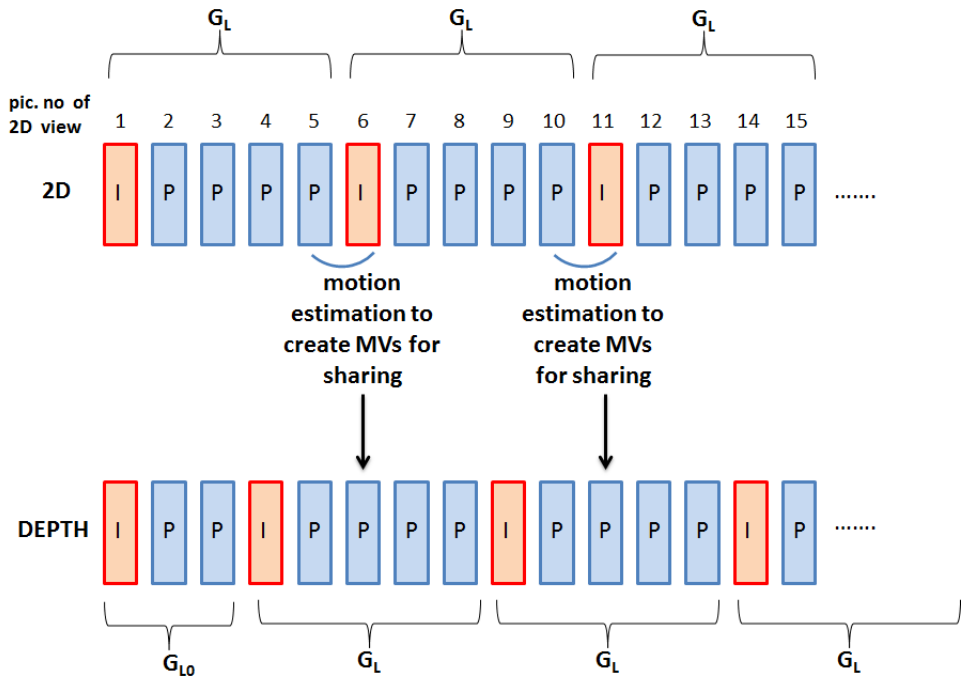


Figure 4.4: Proposed encoding scheme of 2D and depth sequences.

When an entire intra frame loss occurs, conventional error concealment can only utilize limited information. With our proposed encoding scheme, one can conceal a lost 2D I-frame by extracting the MVs from the corresponding inter frame in the depth sequence. Since the MVs are directly inherited from the motion estimation of the 2D intra frame itself, they reflect the motion relation of the lost intra frame and the previous decoded one. Consequently a straightforward motion compensation error concealment (MCEC) without further MV refinement could provide an excellent recovery.

4.3 Experimental Results

The proposed scheme was implemented in H.264/AVC reference software JM15.1. Six 2D+depth sequences are tested: Balloons (512 × 384), Bookarrival (512 × 384), Cafe (640 × 480), Mobile (720 × 480), Newspaper (512 × 384) and Pantomime (640 × 480). The lengths of the sequences are 300, 100, 300, 200, 300,

and 500 frames, respectively. In order to have more intra frames for the generality of the simulation, we successively shift the start frame for encoding so each frame could be intra-coded. For example, in Figure 4.4, the frames with picture number 1, 6 and 11 are intra coded in the 2D sequence. If we shift the start frame by one frame, then the frames with picture number 2, 7 and 12 will be the intra frames. The sequences are encoded at 30 frames per second with "IPPP" GOP structure. The GOP length of both 2D and depth sequences is set to $G_L = 15$ while the first GOP of each depth sequence is shortened to $G_{L0} = 5$ to create non-alignment. The quantization parameters (QPs) are set to 28 for the 2D sequences and 38 for the depth sequences.

The proposed encoding scheme was implemented in two versions: 1) Basic: motion estimation is only done at MB size 16×16 without further partitioning. 2) Advanced: motion estimation is processed at partition sizes from 16×16 to 8×8. For both Basic and Advanced, the intra frame error concealment of the 2D sequence is MCEC with direct MV reuse from its corresponding inter frame of the depth sequence. The methods are denoted Basic and Adv.

Three other methods with different encoding schemes are compared for intra frame error concealment. 1) Zero motion error concealment: use the co-located pixel value of the previous decoded frame to conceal the lost frame. This is used when the 2D and depth sequences are coded conventionally with no offset between GOPs. 2) MCEC with average MVs: though we can not access the motion information from its corresponding frame in the depth view, we could still utilize the MVs from the 2D sequence itself. By gathering the MVs from the previous and next decoded frame, the average of the forward and backward MVs can be deployed. Furthermore, if the depth sequence is coded with the offset GOP but no motion sharing, we could have extra MVs from the depth sequence to assist MCEC. 3) MCEC with median MVs: the median among the MVs from the corresponding inter frame in the depth view, and the forward and backward MVs from the 2D sequence are used. The above three methods are named "Copy", "MCavg" and "MCmed" respectively.

Each intra frame is tested independently, except for the first IDR frame

of the sequences. The average PSNR of all the concealed intra frames and the lossy GOPs are listed in Table 6.2. It shows that our proposed basic and advanced methods significantly outperform the copy and other MCEC algorithms. The proposed basic version can give a PSNR increase from 2.5dB up to 10dB, though the motion compensation is only done unsophisticatedly at 16×16 MB size level. In the case when further partition is deployed for the proposed advanced scheme, an extra $0.7 \sim 1.8$ dB PSNR improvement is attained. By simply having the motion information from the 2D sequence itself, the MCEC based methods easily surpass the copy method. The MCmed method with depth offset encoding only provides a small gain compared to MCavg, since some additional depth motion information is known for MCmed. However, the large improvement appears when the proposed scheme has the combined advantage of frame offset and motion compensation.

Table 4.1: Comparison of the average PSNR performance over all dropped frames and GOPs for different error concealment methods.

Sequence (error free PSNR)	PSNR	Error Concealment Method				
		Copy	MCavg	MCmed	Basic	Adv.
Balloons (40.03)	I	28.58	32.80	32.82	35.96	36.65
	GOP	28.74	32.62	32.61	35.12	35.60
Bookarrival (39.53)	I	26.11	29.05	28.69	33.27	34.89
	GOP	25.09	28.67	27.94	32.97	34.47
Cafe (39.06)	I	32.20	34.87	35.16	36.88	37.67
	GOP	31.94	34.58	34.82	36.49	37.17
Mobile (39.40)	I	30.50	32.35	32.81	35.16	36.00
	GOP	28.85	31.73	32.32	34.65	35.53
Newspaper (38.20)	I	28.21	32.57	32.68	34.66	35.48
	GOP	27.29	31.64	31.60	33.50	34.22
Pantomime (41.55)	I	21.02	26.69	25.34	31.99	33.83
	GOP	21.50	27.52	25.86	32.75	34.32

Since I frames on average require many more bits than P or B frames, the instantaneous output bitrate of a video encoder can be highly uneven. As stated in [55], the frame offset feature of the proposed encoding scheme could help to stabilize the instantaneous bitrate of the coded 2D+depth sequences. The

comparison of the total bitrate for each encoding scheme is shown in Table 4.2. The conventional scheme with aligned 2D and depth GOP is denoted "Orig.", and the frame offset encoding method is denoted "Offset". Our proposed encoding schemes with MV sharing are called "Basic" and "Advanced", based on the partition size level at which motion estimation is deployed. The proposed methods bring a bitrate penalty of $1.3 \sim 2.2\%$, producing a tradeoff between encoding efficiency and error concealment improvement. However, 2D intra frames occupy a large portion of the total 2D+depth bit usage, about 30% in our experimental settings. When network traffic is congested, we could alleviate the situation by dropping some intra frames and incorporating the proposed algorithm at the same time to release more available bandwidth and effectively prevent prominent quality degradation in the case when congestion makes a loss inevitable.

Table 4.2: First column is the total bitrate(kb/s) of the conventional scheme with aligned 2D and depth GOPs. Remaining columns are the percentage increase in bitrate for different encoding schemes.

Sequence	Encoding Scheme			
	Orig.	Offset	Basic	Advanced
Balloons	975.32	0.12	1.34	1.78
Bookarrival	832.82	0.39	1.65	2.00
Cafe	929.45	0.17	1.48	1.74
Mobile	1151.77	0.15	1.54	1.82
Newspaper	856.08	0.16	1.14	1.49
Pantomime	2108.61	0.02	1.91	2.25

4.4 Conclusion

We presented an effective algorithm for recovering whole intra frame loss in 2D+depth video. The error concealment cooperates with a proposed encoding scheme that exploits the motion correlation between the 2D and depth sequences. Experimental results show that our method results in significant PSNR improvement.

Chapter 4 is adapted from Y.-L. Chang and P.C. Cosman, "Depth-Assisted Error Concealment for I-frame loss in 2D+depth Coded Stereoscopic video", submitted to IEEE Signal Processing Letters, 2014. I was the primary author. Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter 4.

Chapter 5

2D+depth video in packet loss environments

In order to combat packet losses over erroneous networks, we often have to pay a bitrate penalty to enhance the robustness of video communications, as does, for example, our work in Chapter 4. One of the approaches for error resilient video coding is to consider both source and channel distortion at the same time and to find a balance between encoding bitrate penalty and error resistance of the transmitted sequences. In this chapter, we propose a motion information sharing encoding scheme with an end-to-end rate-distortion model for H.264/AVC coding of 2D+depth sequences. Experimental results show that the proposed encoding scheme improves PSNR performance for the depth sequence under a packet loss environment without increasing encoding bitrate.

Many techniques have been proposed for error resilient video coding over lossy networks. For some more advanced algorithms, the end-to-end distortion due to compression and packet loss is estimated, and then utilized for mode selection with rate-distortion optimization (RDO) [58–60]. A recursive optimal per-pixel estimate (ROPE) algorithm estimates the pixel level end-to-end distortion by keeping track of the first and second moments of the reconstructed pixel value [58]. An error robust RDO method (ER-RDO), developed for packet-loss environments [59], was adopted in the H.264/AVC test model. ER-RDO estimates the expected overall end-to-end distortion by decoding K random realizations of the lossy channel

at the encoder. This approach could be very accurate if K is large enough, but the computational complexity is extremely high. All of the above work focussed on traditional 2D video. In this chapter, we aim to extend this RDO research to 2D+depth video transmission.

As mentioned in Chapter 4, a depth map contains little texture and predominantly flat patches with sharp edges marking the boundary between objects at different depths, and it is spatially correlated with its corresponding 2D image. In [61], it was shown that direct mode is selected most often in the depth video encoding, but the bits are mostly generated by inter-predicted modes, which take 65% of the overall bitstreams. In addition, motion information takes 59% of the bits in the inter-predicted coding. Conventionally the 2D and depth sequences are independently encoded, and the high similarity between the two sequences is not utilized. However, several works have been done to make use of this correlation [61–65]. In [62], the authors proposed a coding structure for depth map coding with H.264/AVC to share the motion information of the corresponding 2D video by exploiting the similarity of motion vectors between 2D and depth sequences. In [63], motion sharing schemes were implemented in the scalable video coding (SVC) structure and utilized for error concealment; however, the shared MVs were sent repeatedly for both streams and only two encoding modes were used, namely 'macroblock (MB) skip' and 'motion estimation'. As intra mode was not included in R-D optimization, error propagation could be serious under this setting due to the lack of intra refresh. Both [64] and [65] introduced joint motion estimation techniques. In [64], the authors took the means of a joint estimation of the MV field for the texture motion information and depth map sequence while [65] further applied the joint MVs on error concealment, but again, only inter and skip mode were considered for encoding. In addition, none of the above methods included the channel distortion for encoding mode selection in lossy networks.

We would like to address the problem of transmitting 2D+depth sequences over error-prone networks using rate distortion optimized mode selection in this chapter. The chapter is organized as follows: In Section 5.1, the proposed algorithm is described. Section 5.2 presents the experimental results and discussions, while

Section 5.3 summarizes our conclusions.

5.1 Overview of the Proposed Method

5.1.1 End-to-End Distortion Model

Video standards such as H.264/AVC provide various intra and inter modes to encode a MB. In order to select the best mode for each MB, a Lagrangian optimization technique is used to minimize the distortion subject to a rate constraint [66]. Based on the following equation, the coding mode that minimizes the Lagrangian cost is chosen to code the macroblock m in frame n,

$$\min_{mode}(J(n, m, mode)) = \min_{mode}(D(n, m, mode) + \lambda R) \tag{5.1}$$

where λ is the Lagrangian multiplier for the mode decision given by $\lambda = 0.85 \times 2^{(QP-12)/3}$ in H.264/AVC. R denotes the number of bits needed for coding the MB in the specified mode, which includes the bits for the MB header, motion vector, reference frame, and transformed coefficients. $D(n, m, mode)$ represents the distortion of the MB.

In [60], the authors used an end-to-end distortion for mode selection that consists of source, error-propagated, and error concealment distortions. Suppose p is the packet loss rate, REF lists the reference frames and m_J lists the motion vectors of all subblocks in macroblock m in frame n in terms of coding option $mode$. The end-to-end distortion is:

$$D(n, m, mode) = (1 - p)(D_s(n, m, mode) + D_{ep}(REF, m_J)) + pD_{ec}(n, m) \tag{5.2}$$

where $D_s(n, m, mode)$, $D_{ep}(REF, m_J)$ and $D_{ec}(n, m)$ denote the macroblock-level source distortion, error-propagated distortion, and error concealment distortion respectively. The error propagated distortion D_{ep} can be recursively calculated after the current frame has been encoded, and stored as a distortion map for further reference. The new Lagrange multiplier in a packet loss environment was also derived as $(1-p)\lambda$. Since $D_{ec}(n, m)$ is independent of $mode$, it is unnecessary

to calculate for the mode selection. Therefore, the final formula for the Lagrangian cost is:

$$\begin{aligned} J(n,m,mode) &= (1-p)(D_s(n,m,mode) + D_{ep}(REF, m_J)) + (1-p)\lambda R \\ &= (D_s(n,m,mode) + D_{ep}(REF, m_J)) + \lambda R \end{aligned} \tag{5.3}$$

5.1.2 Proposed Motion-Sharing Encoding Scheme

The proposed encoding scheme of video-plus-depth sequences is illustrated in Fig 5.1. The 2D sequence is encoded first, and the intra or inter prediction is performed as in the conventional H.264/AVC, which does the motion compensation prediction (MCP) between each frame. We also employ the end-to-end distortion model in Sec. 5.1.1 for the mode selection.

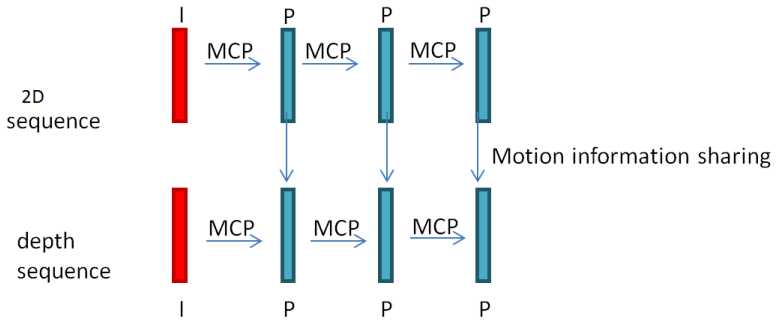


Figure 5.1: Encoding scheme of 2D+depth sequences

For the depth sequence, one extra mode is introduced to the mode selection process, which is the motion information sharing mode. After encoding the 2D sequence, if inter-prediction mode is selected for a certain MB, its motion information will be passed to the corresponding MB in the depth sequence as a candidate MV for mode selection. In the motion information sharing mode, we do not need any bits to represent the MV since it is shared from the texture sequence. The end-to-end distortion model is used for mode selection for the depth sequence also, with the addition of the motion sharing mode.

Moreover, in [60], for a lost pixel in frame n, the error concealment is defined as copying pixels from frame $n-1$. Given that we could have motion vectors

from both texture and depth sequences, for a lost MB in the depth sequence, the error concealment is ameliorated by gathering the MVs from the surrounding MBs and the corresponding MB from the texture sequence, and then using boundary matching to find the best MV for motion compensation. Under the consideration of the end-to-end distortion model with the extra mode and the ameliorated concealment, the best mode will be selected from intra, inter, skip and motion sharing.

5.2 Experimental Results

The proposed algorithm was implemented in H.264/AVC reference software JM15.1. The error-resilient video coding algorithm proposed in [60] is taken as a reference in the comparison. Three 2D+depth sequences are used for experiments, namely Cafe, Dancer, and Balloons. In our experiments, we focus the performance on the depth sequence. These three depth sequences represent different types of depth maps. The depth maps of Cafe and Dancer both have smooth edges; the former is calculated by a state-of-the-art stereo matching algorithm from multiple views, while the latter one is the ground truth from computer graphics. The depth map of Balloons is also calculated but with very coarse boundaries. The difference between smooth and coarse depth maps is shown in Figure 5.2.

(a) Dancer: 2D image

(b) Dancer: depth map

(c) Balloons: 2D image

(d) Balloons: depth map

Figure 5.2: Examples of different types of depth maps; (a) 2D image from Dancer sequence and (b) corresponding depth map: smooth edge, ground truth. (c) 2D image from Balloons sequence and (d) corresponding depth map: coarse boundary, calculated.

The sequences are encoded at 30 frames per second (fps) for 100 frames, and only the first frame is encoded as an I frame and the remaining frames are encoded as P frames. Each row of macroblocks composes a slice and is transmitted in a separate packet. Hence each packet is independently decodable. We assume that the first frame is conveyed reliably. The packet loss situation is simulated according to the error resilience testing condition specified in [67]. The packet loss rates (PLR) at 0%, 5%, 10%, 15% and 20% are tested for 100 lossy realizations on each.

Distributions of Coding Modes with Packet loss:

The percentage of time that a coding mode is optimal is determined by its R-D behavior. In Tables 5.1 and 5.2, we present these distributions. Table 5.1 shows the distribution of the conventional encoding method, which is without motion information sharing, and Table 5.2 presents the result with the new encoding scheme. For depth map encoding, these results show that skip mode is dominant due to the nature of the simple content. For both methods, intra mode is selected the least, and these intra-coded MBs are mostly located at the edges of objects. Comparing the two tables, we observe that motion sharing mode mostly replaces inter mode in the proposed encoding scheme. For higher packet loss rates, the usage of intra mode increases while the motion sharing mode is used less.

Table 5.1: Distribution of various coding modes in packet loss environments (%) without motion sharing- Cafe depth sequence

Loss Rate	Skip	Inter	Intra
0%	85.43	14.41	0.15
5%	85.55	13.01	1.43
10%	85.82	12.04	2.13
15%	86.27	11.07	2.65
20%	86.45	10.40	3.19

Table 5.2: Distribution of various coding modes in packet loss environments (%) with motion sharing- Cafe depth sequence

Loss Rate	Share	Skip	Inter	Intra
0%	7.88	84.30	7.68	0.13
5%	5.50	85.02	8.14	1.34
10%	4.73	85.49	7.80	1.98
15%	4.21	85.41	7.81	2.56
20%	3.35	85.80	7.71	3.12

Performance Evaluation for the Proposed Scheme:

We assume that the texture sequence is transmitted correctly and the depth sequence is transmitted with packet loss. Under this condition, error propagation only comes from the depth stream itself. Figures 5.3, 5.4 and 5.5 show the PSNR performance results of the proposed and reference methods under the given condition. In Figures 5.3 and 5.4, the proposed method with motion sharing mode has 0.2-1.0 dB performance improvement over the reference scheme. In Figure 5.5, the proposed method is slightly worse than the reference scheme. The degradation is caused by the nature of the Balloons depth map. As mentioned before, the Balloons depth map has coarse boundaries due to the depth calculation method it employed. This decreases the advantage of the motion sharing mode since object edges in the depth map will not be accurately aligned with those in the 2D image. Consequently the selected mode will not be resilient to losses.

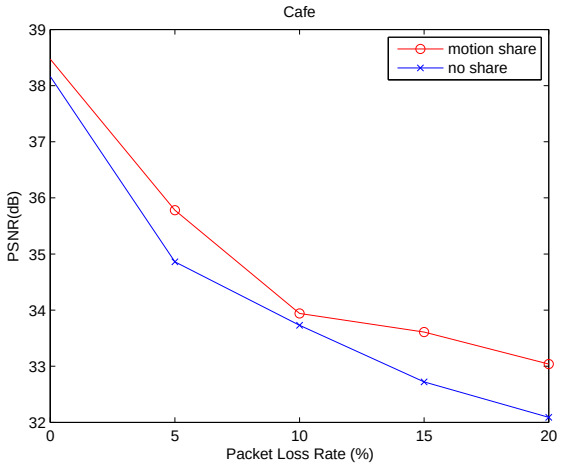


Figure 5.3: Average PSNR(dB) performance comparison of proposed and reference methods, Cafe (320×240), 64kbps

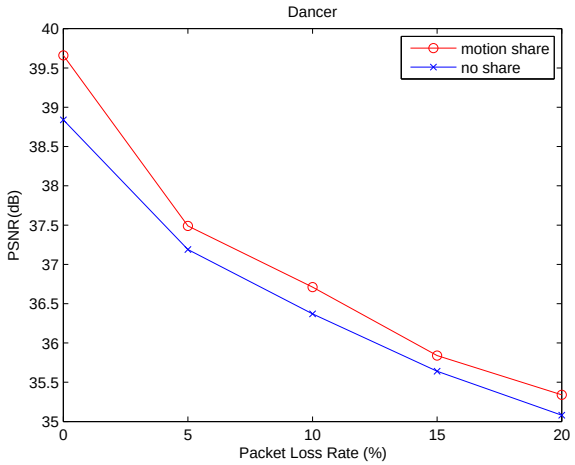


Figure 5.4: Average PSNR(dB) performance comparison of proposed and reference methods, Dancer (480×272), 64kbps

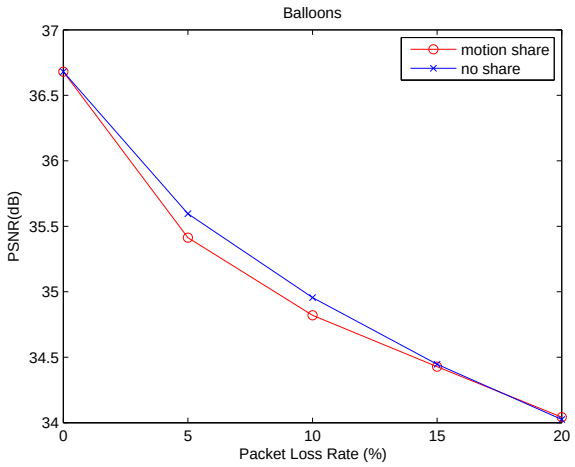


Figure 5.5: Average PSNR(dB) performance comparison of proposed and reference methods, Balloons (512×384), 96kbps

5.3 Conclusion

We present a novel method for coding the 2D+depth sequences by introducing an extra motion sharing mode to the depth stream. The extra mode is generated by utilizing the MV from the texture stream. By sharing the MV, we not only save bits but also make use of the shared MV as error concealment. The

mode selection process is implemented based on the estimation of end-to-end R-D cost. For sequences with precise depth calculation, our proposed method achieves a better PSNR performance in packet loss environments.

Chapter 5 is adapted from Y.-L. Chang, Y. Zhang and P.C. Cosman, "Joint Source-Channel Rate-Distortion Optimization with Motion Information Sharing for H.264/AVC Video-plus-Depth Voding" submitted to Asilomar Conference on Signals, Systems and Computers, 2014. I was the primary author. Co-author Prof. Cosman and Prof. Zhang directed and supervised the research which forms the basis for Chapter 5.

Chapter 6

Motion compensated error concealment for HEVC based on block-merging and residual energy

High Efficiency Video Coding (HEVC) is the latest coding standard in 2013. Though the main goal of the HEVC standardization effort is to enable significantly improved compression performance, up to 50% bitrate reduction, relative to existing standards for equal perceptual video quality, error recovery for HEVC has not been addressed yet. In this chapter, we propose a motion-compensated error concealment for HEVC that can preserve edge and object structure information without involving motion estimation or object detection at the decoder. First, residual energy of each block is analyzed to determine the reliability of each MV. Instead of refining the motion field by further partitioning each block into smaller blocks, we merge adjacent blocks that have unreliable MVs into a larger region. The merged block is assigned one single motion vector. Since the blocks with unreliable motion vectors are concealed using the same motion vector, the edges and the structure of the objects can be kept.

The chapter is organized as follows: In Section 6.1, an overview of HEVC is presented. In Section 6.2, we present the proposed algorithm in detail. The experimental results are demonstrated in Section 6.3. Section 6.4 summarizes our conclusions.

6.1 Overview of HEVC

High Efficiency Video Coding (HEVC) is the latest video coding technology standard. As a joint project of the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC Moving Picture Experts Group (MPEG), the Joint Collaborative Team on Video Coding (JCT-VC) was formed for HEVC development, and the standard was approved and formally published in 2013. As a successor to H.264/AVC, HEVC promises to reduce the overall cost of delivering and storing video assets without decreasing the quality of experience delivered to the viewer, and two key issues have been a particular focus: increased video resolution and increased use of parallel processing architectures [68].

Though HEVC is designed to achieve multiple goals, including coding efficiency and ease of transport system integration and data loss resilience, it provides no guarantees of end-to-end reproduction quality and does not suggest any concealment when the bitstream is lossy. Various error concealment methods have been proposed to overcome packet loss in video transmission for prior standards [9]. In [69] and [70], the pixel values were recovered by spatially interpolating available pixels in neighboring macroblocks (MBs). The boundary matching algorithm (BMA) [57] and decoder motion vector estimation (DMVE) estimated lost motion vectors (MVs) based on taking the candidate MVs from its spatial and temporal neighbors, by minimizing a given distortion measure between the correctly received pixels. These spatial and temporal error concealment schemes were addressed to the coding characteristics of the MB-based codec by exploiting the correlation between a damaged MB and its adjacent ones in the same or previous frame. However, though HEVC is still under the block-based motion-compensation and transform coding structure, there is no MB in the codec. The macroblock concept has been extended by defining three types of variable size unit: Coding Unit (CU), Prediction Unit (PU) and Transform Unit (TU).

Starting from the largest CU (LCU), each CU allows recursive quadtree splitting into multiple sub-CUs, for sizes from 64×64 (CU depth=0) to 8×8 (CU depth=3). Each sub-CU can be further split into multiple PUs. PUs are the basic unit for motion prediction. After PU segmentation, a proper size of TU

is determined for residual coding. Two encoding modes are supported in HEVC: intra- and inter-picture prediction. The mode is specified at the CU level, meaning all the PUs in a CU will be predicted under the same mode. In HEVC, a slice is the data structure that can be delivered and decoded independently, and it can either be an entire frame or a region of a frame. Slices are a sequence of LCUs, while they are composed of MBs in the prior standards. The hierarchical decision level for HEVC is shown in Figure 6.1. LCUs are usually set to be the size of 64×64, which is sixteen times larger than a 16×16 MB in the prior standards, thus slice losses from a HEVC bitstream will generally involve a vaster area of a frame. Under this condition, many of the prior error concealment methods would not be applicable, since they were usually designed for smaller lost blocks and often took the nearby correctly received pixels as reference. Because a loss in HEVC contains at least one LCU, the large lost block makes most of the lost pixels distant from the correctly received pixel border, thus the distortion measure between the block edge no longer serves as a good criterion for recovery.

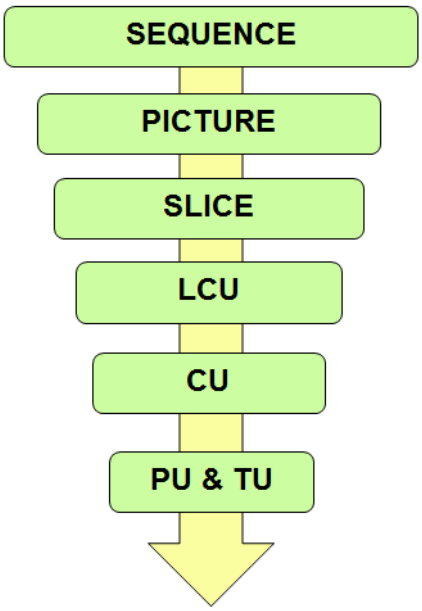


Figure 6.1: Hierarchical decision level for HEVC.

Few studies have been done regarding error concealment in HEVC. In [71], a motion vector extrapolation based method was proposed for whole frame loss. MV correlation from the co-located LCU was calculated for deciding whether to divide a large block into smaller ones or not. However, in block-based motion estimation at the encoder, the motion vector field is generated by minimizing the energy of prediction residuals and the rate-distortion cost, which may make those estimated MVs fail to represent the true motion [72–74], thus it could be improper to take every MV from the co-located LCU. In addition, a loss is not necessarily a whole frame loss and could be partial, which we call a slice loss in this chapter. For slice loss, spatial misalignment is more likely to happen and degrades the video quality.

Currently in HEVC reference software HM [75], only frame level concealment is implemented, where pixel copy from the previous frame is used. A slice loss is not yet detected and concealed. Among all types of error concealment, motion trajectory reuse from the co-located LCUs will be a relevant approach since the coding structure of HM preserves the information from the adjacent and co-located LCUs. In this chapter, we propose a motion-compensated error concealment that can preserve edge and object structure information without involving motion estimation or object detection at the decoder. First, residual energy of each block is analyzed to determine the reliability of each MV. Instead of refining the motion field by further partitioning each block into smaller blocks, we merge adjacent blocks that have unreliable MVs into a larger region. The merged block is assigned one single motion vector. Since the blocks with unreliable motion vectors are concealed using the same motion vector, the edges and the structure of the objects can be kept.

6.2 Proposed method

In this section, we propose a motion-compensated error concealment scheme based on the classification map of residual energy associated with each motion vector, and based on a block-merging algorithm.

Although network applications are one of the targets of HEVC, HEVC has not yet addressed transmission in networks other than to mandate byte stream compliance with Annex B of H.264/AVC. In [76], a streaming framework is designed and implemented. However, it used pre-generated encoder trace files and receiver trace files to detect loss, which is not a very realistic approach. The current HM so far is not able to detect and conceal a slice loss. However, a syntax element in HEVC, slice_segment_address, specifies the address of the first LCU in the slice segment, in a coding tree block raster scan of a frame. By modifying HM and tracking this syntax, we could detect a slice loss without auxiliary files. If the correctly decoded LCU number in a frame is discontinuous with the next slice_segment_address value in the same frame, a slice loss is detected.

Fig. 6.2 shows the block diagram of the proposed method. Each lost LCU will be concealed sequentially.

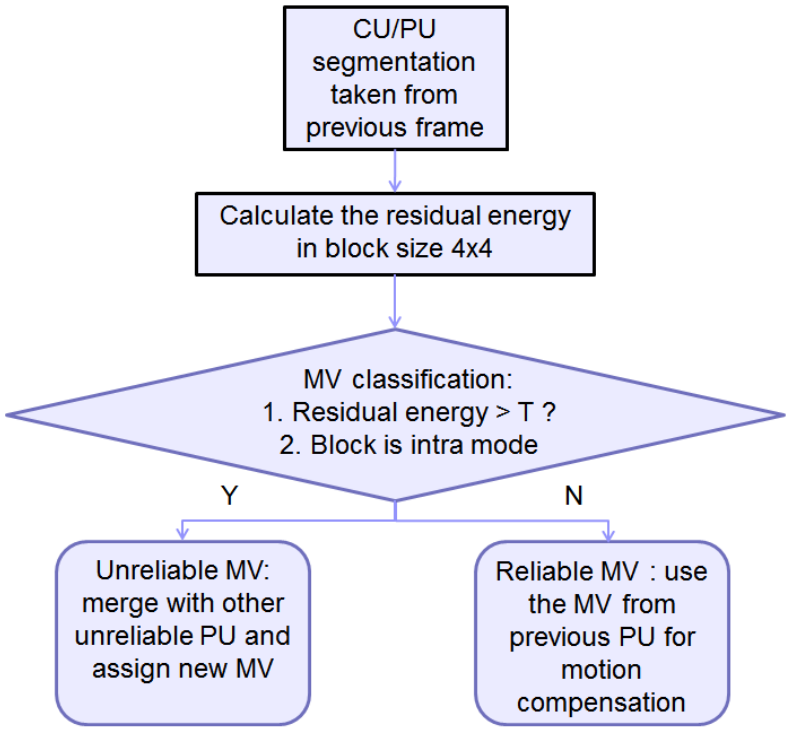


Figure 6.2: Block diagram of the proposed method.

Step 1: CU and PU segmentation:

For a lost LCU, its CU/PU partition information is lost. In [77, 78], the authors showed that much of the time, the CU depth is highly correlated with its co-located one in the previous frame, and so is PU segmentation. In the first step of the block diagram, the algorithm assumes the lost LCU has the same CU partition and PU segmentation as the co-located one.

Step 2: Motion vector classification based on residual energy:

As HEVC still uses block-based motion estimation, it is possible that an MV is selected for reasons of rate-distortion efficiency rather than because it represents the true motion. Therefore, when a PU has high residual energy, we can reasonably argue that the MV may not be reliable for representing the true motion. In our proposed method, we assume that the motion of the lost CU will follow the same trajectory as its co-located one, so the motion vector field from the co-located CU will be utilized for error concealment. In the second step in Figure 6.2, to classify the candidate MVs from the co-located CU as reliable or not, the residual energy, E, of the co-located CU is calculated for each 4×4 block, $b_{m,n}$, by taking the sum of the absolute value of the luma reconstructed prediction error for each pixel.

$$E = \sum_{(i,j) \in b_{m,n}} | \, r_Y(i,j) \, |$$

$r_Y(i,j)$ is the reconstructed residual signal of the Y component. If E is smaller than a threshold, the 4×4 block is classified as a reliable region, otherwise it is classified as unreliable. The threshold is selected from a heuristic search. In addition, an intra CU will also be categorized as unreliable. If any 4×4 block in a PU is unreliable, the whole PU is signaled as an unreliable PU, thus the corresponding MV of this PU is also unreliable. Fig. 6.3 demonstrates an example: an 8×8 CU is divided vertically into two 4×8 PUs, and there are two 4×4 residual blocks in each PU. The yellow denotes an unreliable block while blue stands for reliable ones. For the right PU, one of the residual block is unreliable, so the whole PU is defined as unreliable.

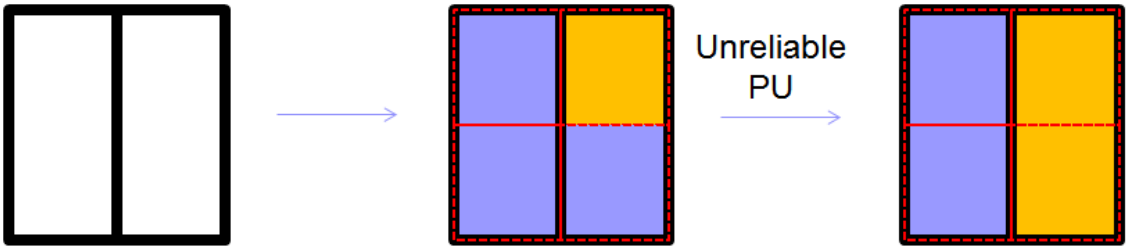


Figure 6.3: Example of unreliable PU classification.

Step 3: PU merging and MV reassignment:

In the last step of Figure 6.2, unreliable PUs will be merged and reassigned with refined MVs. When there is a group of adjacent PUs that have unreliable motion vectors, misalignment happens easily along the edges, and the shape of the objects usually could not be maintained. To keep the integrity of the object, it would be beneficial to group these units with one MV, so the structure would not be deformed. The merging process only happens between PUs in the same CU depth, so the merged PUs would not be too different in size. This reduces the blockiness effect. There is no merging at CU size 32×32 and 64×64 to refrain from losing too much detailed motion. Starting from the very top left point of a lost region, whenever we encounter an unreliable PU, we check its right, bottom, and bottom-right PUs. If any of these PUs are both unreliable and within the same CU level, they are merged into a larger piece. For each PU, it will only be merged once to keep the size of the group in a reasonable range. Figure 6.4 depicts an example: Four 16×16 CUs contain five PUs. Three of the PUs are unreliable, so they are merged into one piece.

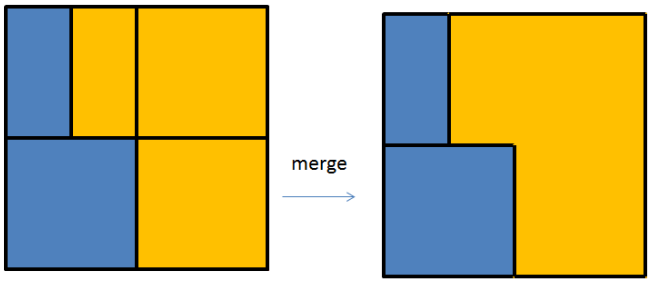


Figure 6.4: Example of merging unreliable PUs.

After PU merging, the reliable MVs from its adjacent PUs are collected, and the average of these reliable MVs will be assigned. For a reliable PU, the MV from the co-located area will be used directly.

6.3 Simulation

In this section, we present experimental results to evaluate the proformance of the proposed method. The proposed method will be compared with two other schemes:

1. Pixel copy

2. Basic motion compensated error concealment (MCEC), where the MV from the co-located CU is applied directly with no refinement. For an intra CU, pixel copy is used.

Two video sequences, Soccer (720×480) and Drill (832×480) , are encoded by HM11.0 for 120 frames. The frame rate is 30 frame per second (fps) for Soccer and 50 fps for Drill; the intra period is every 20 frames with only P frames in between. The QP is 28 and each slice has a fixed 20 LCUs to simulate the loss of a region of a frame.

The proposed algorithm is implemented in HM10.0. We use the packet loss simulator developed by AHG14 to facilitate G.1050/TIA 921 packet loss simulations for HEVC [79]. The loss is randomly distributed in all the P frames. The

packet loss rates (PLRs) of 1%, 3%, 5%, and 10% are tested, and each sequence is decoded for 100 random realizations.

Table 6.1 presents the average PSNR of the first erroneous frame in each GOP, and hence the influence of the error propagation is excluded; also the PSNR value does not decrease with higher PLR. As shown in the table, the proposed method yields higher PSNR than the copy and MCEC algorithm by up to 1.2 dB. Table 6.2 presents the PSNR performances averaged over all frames for different sequences with different PLRs. As shown in the table, the proposed method outperforms the copy and MCEC algorithm up to 0.26 dB. Since the error propagation of HEVC is quite severe and the quality of succeeding frames degrades very fast for all types of concealment methods, the gain of the proposed method is not much in terms of the PSNR over the whole sequence. However, if we only look at the erroneous frame itself, the PSNR gain is much prominent and the visual quality is also better. In both Tables 6.2 and 6.1, MCEC performs the worst because the improper reuse of the co-located MVs makes the concealed area quite blocky. The copy method roughly maintain the shape of the object but fails at the boundary of the corrupted area.

Table 6.1: Comparison of the average PSNR performance over the first erroneous frame in each GOP for different PLRs

Sequence	Method	Packet Loss Rate			
		1%	3%	5%	10%
Soccer	copy	28.63	28.93	28.80	28.99
	MCEC	28.45	28.79	28.71	28.91
	proposed	29.82	30.00	29.82	29.63
Drill	copy	31.31	30.96	31.05	30.60
	MCEC	31.08	30.75	30.92	30.50
	proposed	32.51	32.22	32.21	31.5

The visual comparisons are presented in Figures 6.5 and 6.6, demonstrating examples of frame 87 of the Soccer sequence and frame 24 of the Drill sequence, where (a) is the original compressed frame without loss, (b) is the corrupted frame and (c)-(e) are the frames concealed using copy, MCEC, and the proposed algo-

Table 6.2: Comparison of the average PSNR performance over all frames for different PLRs

Sequence	Method	Packet Loss Rate			
		1%	3%	5%	10%
Soccer	copy	28.63	24.58	22.07	19.84
	MCEC	28.56	24.51	21.99	19.77
	proposed	28.84	24.74	22.28	20.02
Drill	copy	30.93	27.05	25.32	23.13
	MCEC	30.86	26.97	25.25	23.06
	proposed	31.17	27.31	25.58	23.34

rithm respectively. In both figures, the visual quality of our method is significantly better than the others. Our method successfully preserves the shape of the moving objects with smooth edges while the copy and MCEC methods fail to maintain the structure of moving objects with blockiness and deformed boundary. Comparing with two other methods, the PSNR value of the proposed method is up to 2.9 dB higher for Soccer and 4.5 dB higher for Drill in these cases. This again proves the effectiveness of our algorithm.

(a) original

(b) corrupted

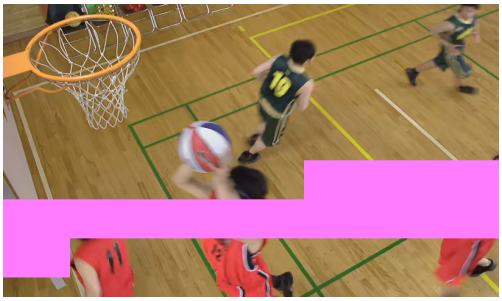
(c) copy

(d) MCEC

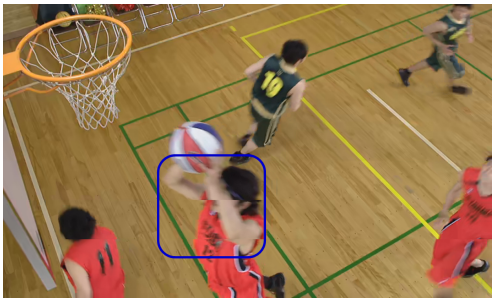
(d) proposed

Figure 6.5: Reconstructed results of frame 87 of the Soccer sequence: (a) original frame, (b) corrupted frame, (c) concealed by copy, PSNR: 22.58dB (d) concealed by MCEC, PSNR: 22.57dB (e) concealed by the proposed method, PSNR: 27.16dB.

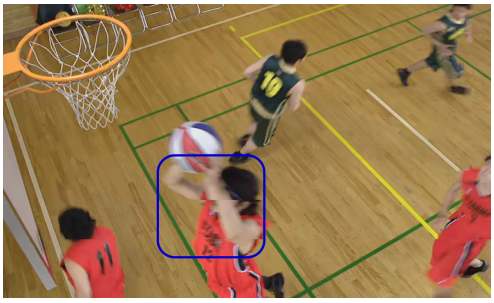
(a) original



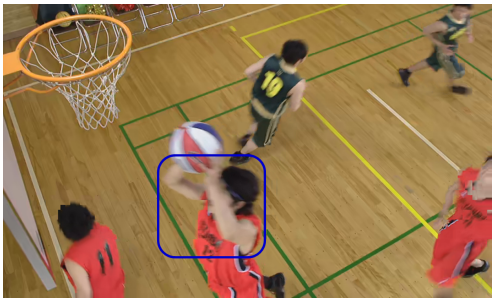
(b) corrupted



(c) copy



(d) MCEC



(e) proposed

Figure 6.6: Reconstructed results of frame 24 of the Drill sequence: (a) original frame, (b) corrupted frame, (c) concealed by copy, PSNR: 30.62dB (d) concealed by MCEC, PSNR: 30.54dB (e) concealed by the proposed method, PSNR: 33.58dB.

6.4 Conclusion

We propose a motion-compensated error concealment method for HEVC and implement the method in reference software HM. Based on the received residual information, the motion vector reliability is analyzed and classified. The CU with unreliable MVs will be merged and assigned with one new MV to maintain the structure of the moving object and edge information. Our method is effective yet simple without doing edge or object detection explicitly. Though the achieved gain in terms of PSNR appears marginal, the improvement of visual quality is prominent.

Chapter 6 of this dissertation, in part, is a reprint of the material as it appears in Y.-L. Chang, Y. Reznik, Z. Chen, P.C. Cosman, "Motion Compensated Error Concealment for HEVC Based on Block-Merging and Residual Energy," IEEE Packet Video Workshop, 2013. I was the primary author and the co-authors Prof. Cosman, Dr. Reznik and Dr. Chen directed and supervised the research which forms the basis for Chapter 6.

Chapter 7

Conclusion

In this dissertation, we have proposed a network-based packet loss importance model, a network-based whole frame loss visibility model and several error concealment methods. We deploy these approaches either in the networks or at the decoder side to improve the visual quality of end-users.

In Chapter 2, we develop a network-based model for fixed-sized IP packets to predict visual importance using an objective experiment. Our results show that, for a fixed-sized IP packet, the most significant factors in the model are frame type, and temporal and spatial location. Since the packets are fixed size in bytes, a P-packet covers a much larger pixel area than an I-packet, and so causes more quality degradation when lost.

In Chapter 3, we develop a network-based packet loss visibility model for whole frame loss. We present a subjective experiment and its results on whole B frame loss visibility for H.264/AVC encoded bitstreams. We examine the visual effect of whole frame loss by different decoders. We find that about 39% of whole frame losses of B frames are not observed by any of the subjects, and over 58% of the B frame losses are observed by 20% or fewer of the subjects. Using simple predictive features which can be calculated inside a network node with no access to the original video and no pixel level reconstruction of the frame, we developed models which can predict the visibility of whole B frame losses. The models are then used in a router to predict the visual impact of a frame loss and perform intelligent frame dropping to relieve network congestion. Dropping frames based

on their visual scores proves superior to random dropping of B frames.

In Chapter 4, we develop an error concealment method with a refined 2D+depth encoding scheme to combat whole intra frame loss in the 2D sequence. The frame offset encoding technique is deployed with motion vector sharing between the 2D and depth sequences so that the inter frame in the depth stream has MVs that describe the scene in its corresponding 2D frame more precisely. These accurate MVs will assist the recovery of an entire intra frame loss by a simple motion compensation error concealment. The proposed algorithm has significant PSNR improvement in our simulation with little bitrate penalty.

In Chapter 5, we develop a joint source-channel coding scheme for 2D+depth video format, to extend the encoding modes for depth sequences based on an end-to-end distortion model. We first add an extra motion information sharing mode for the depth sequence, and then improve the error concealment methods. Based on these changes, we use a distortion model that considers both encoding and channel distortion for Rate-Distortion optimized video-plus-depth mode selection in packet-loss environments by taking account of the network conditions, i.e. the packet loss rate. Experimental results with the proposed encoding scheme show PSNR gains of up to 1 dB for the depth sequence under a packet loss environment.

In Chapter 6, we develop a motion-compensated error concealment method for HEVC and implement the method in reference software HM. The motion vector from the co-located block will be refined for motion compensation. Based on the reliability of these MVs, blocks will be merged and assigned with new MVs. The experimental result shows that not only the visual quality performs well but also a substantial PSNR gain.

7.1 Future work

The future work related to packet importance modeling and video transmission for improving end-user perceptual quality includes:

- *Improving error resilience for HEVC:* Since HEVC achieves higher compression factors, on average each bit in HEVC contains more information than

prior codecs. This causes HEVC to have more serious error propagation when packets are lost. This drawback could be compensated by employing better channel coding. A joint source and channel coding can help to increase its error robustness and to minimize the bitrate increase at the same time.

- *Versatile packet loss experiment for HEVC:* HEVC can be transmitted in either a slice format or a tile format. A slice refers to a horizontally scanned area in a frame, while a tile refers to a rectangular region in a frame. The tile design benefits parallel hardware implementation. However it could bring a quite different visual impact to end-users when a tile is lost. Conducting a packet loss experiment for different packetization to evaluate the lossy effect of slices and tiles is an important goal.

- *Network-based packet loss visibility model for HEVC:* Since the HEVC standard was just finalized in 2013 with many different encoding features from prior standards, its perceptual quality performance in the presence of packet losses is not yet well understood. A subjective experiment for packet loss visibility in HEVC is a direction to extend our packet loss importance modeling work. The data from the experiment can help us analyze the lossy effect for HEVC and can be used to build a visibility model. As a large part of the HEVC design goal is to focus on low-delay applications, a network-based model will be our next approach.

Bibliography

[1] Y. Wang, "Survey of objective video quality measurements," *EMC Corporation Hopkinton, MA*, vol. 1748, pp. 39, 2006.

[2] M. H. Pinson and S. Wolf, "A new standardized method for objectively measuring video quality," *IEEE Transactions on Broadcasting*, vol. 50, no. 3, pp. 312–322, 2004.

[3] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002.

[4] M. Ghanbari and V. Seferidis, "Cell-loss concealment in atm video codecs," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 3, no. 3, pp. 238–247, 1993.

[5] P. Salama, E. J. Delp, N. B. Shroff, and E. J. Coyle, "Error concealment techniques for encoded video streams," in *International Conference on Image Processing*. IEEE Computer Society, 1995, vol. 1, pp. 9–9.

[6] S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," in *IEEE International Conference on Communications, ICC'95*. IEEE, 1995, vol. 3, pp. 1778–1783.

[7] P. Cuenca, L. Orozco-Barbosa, A. Garrido, F. Quiles, and T. Olivares, "A survey of error concealment schemes for MPEG-2 video communications over ATM networks," in *Canadian Conference on Electrical and Computer Engineering, Engineering Innovation: Voyage of Discovery.* IEEE, 1997, vol. 1, pp. 118–121.

[8] J.-W. Suh and Y.-S. Ho, "Error concealment based on directional interpolation," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 3, pp. 295–302, 1997.

[9] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, 1998.

[10] S. Cei and P. C. Cosman, "Comparison of error concealment strategies for MPEG video," in *Wireless Communications and Networking Conference, WCNC*. IEEE, 1999, pp. 329–333.

[11] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. C. Cosman, and A. R. Reibman, "A versatile model for packet loss visibility and its application to packet prioritization," *IEEE Transactions on Image Processing*, vol. 19, no. 3, pp. 722–735, 2010.

[12] T.-L. Lin and P. C. Cosman, "Efficient optimal RCPC code rate allocation with packet discarding for pre-encoded compressed video," *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 505–508, 2010.

[13] T.-L. Lin and P. C. Cosman, "Network-based packet loss visibility model for SDTV and HDTV for H.264 videos," in *International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE, 2010, pp. 906–909.

[14] "ITU-R BT.710-4 Subjective Assessment Methods for Image Quality in High-Definition Television," Jan 1998.

[15] "DSL Forum Technical Report TR-126: Triple-play Services Quality of Experience (QoE) Requirements," Dec 2006.

[16] The official website of FFMPEG: http://ffmpeg.org/.

[17] T.-L. Lin, J. Shin, and P. C. Cosman, "Packet dropping for widely varying bit reduction rates using a network-based packet loss visibility model," in *Data Compression Conference (DCC), 2010*. IEEE, 2010, pp. 445–454.

[18] also known as ITU-T Rec. H.222.0 ISO/IEC standard 13818-1.

[19] The website for tsMuxeR software: http://www.smlabs.net/tsmuxer_en.html.

[20] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 645–656, 2003.

[21] P. McCullagh and J. A. Nelder, *Generalized Linear Models, 2nd ed.*, Chapman & Hall, 1989.

[22] D. W. Howmer and S. Lemeshow, *Applied Logistic Regression, 2nd ed.*, Wiley-Interscience, 2000.

[23] G. M. Mullet, "Why regression coefficients have the wrong sign," *Journal of Quality Technology*, vol. 8, no. 3, 1976.

[24] A. R. Reibman, V. Vaishampayan, and Y. Sermadevi, "Quality monitoring of video over a packet network," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 327–334, Apr 2004.

[25] S. Tao, J. Apostolopoulos, and R. Guerin, "Real-Time Monitoring of Video Quality in IP Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 5, pp. 1052 – 1065, Oct. 2008.

[26] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-Reference Video Quality Monitoring for H.264/AVC Coded Video," *IEEE Transactions on Multimedia*, vol. 11, no. 5, pp. 932 – 946, Aug. 2009.

[27] A. Eden, "No-Reference Estimation of The Coding PSNR for H.264-coded Sequences," *Consumer Electronics, IEEE Transactions on*, vol. 53, pp. 667 – 674, May 2007.

[28] B. Girod, *What's wrong with mean-squared error?*, MIT Press, Cambridge, MA, USA, 1993.

[29] C. J. Hughes, M. Ghanbari, D. E. Pearson, V. Seferidis, and J. Xiong, "Modeling and subjective assessment of cell discard in ATM video," *IEEE Transactions on Image Processing*, vol. 2, no. 2, pp. 212–222, April 1993.

[30] K. Yamagishi and T. Hayashi, "Parametric Packet-Layer Model for Monitoring Video Quality of IPTV Services," *IEEE International Conference on Communications*, 2008.

[31] N. Montard and P. Bretillon, "Objective Quality Monitoring Issues in Digital Broadcasting Networks," *IEEE Transactions on Broadcasting*, 2005.

[32] J. Hu and H. Wildfeuer, "Use of content complexity factors in video over ip quality monitoring," *International Workshop on Quality of Multimedia Experience*, 2009.

[33] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Transactions on Multimedia*, vol. 8, pp. 207 – 218, April 2006.

[34] N. Staelens, B. Vermeulen, S. Moens, J.-F. Macq, P. Lambert, R. Van de Walle, and P. Demeester, "Assessing the influence of packet loss and frame freezes on the perceptual quality of full length movies," *International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*, 2009.

[35] The website for VQM software: http://www.its.bldrdoc.gov/resources/video-quality-research/software.aspx.

[36] H. Himmanen, M. M. Hannuksela, T. Kurki, and J. Isoaho, "Objectives for new error criteria for mobile broadcasting of streaming audiovisual services," *EURASIP Journal on Advances in Signal Processing*, 2008.

[37] M. H. Loke, E. P. Ong, W. Lin, Z. Lu, and S. Yao, "Comparison of video quality metrics on multimedia videos," *IEEE ICIP*, October 2006.

[38] "Alcatel-Lucent Technical Paper : Access Network Enhancements for the Delivery of Video Services," May 2005.

[39] Q. Huynh-Thu and M. Ghanbari, "Impact of Jitter and Jerkiness on Perceived Video Quality," *International Workshop on Video Processing for Consumer Electronics*, 2006.

[40] R. Pastrana-Vidal, J. Gicquel, C. Colomes, and H. Cherifi, "Sporadic frame dropping impact on quality perception," *Proceedings of the SPIE Human Vision and Electronic Imaging*, vol. 5292, pp. 182–193, 2004.

[41] R. Pastrana-Vidal and J. Gicquel, "Automatic quality assessment of video fluidity impairments using a no-reference metric," *Intl Workshop on Video Proc. and Quality Metrics*, Jan. 2006.

[42] H.264/AVC JM Software : http://iphome.hhi.de/suehring/tml/.

[43] R. Larsen and M. Marx, *An Introduction to Mathematical Statistics and Its Applications*, Pearson Edu, 4th edition.

[44] K. Muller, P. Merkle, G. Tech, and T. Wiegand, "3D video formats and coding methods," in *17th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2010, pp. 2389–2392.

[45] ITU-T and ISO/IEC JTC 1, "Advanced video coding for generic audiovisual services," .

[46] ISO/IEC JTC1/SC29/WG11, "Text of ISO/IEC 14496- 10:200X/FDAM 1 multiview video coding," July 2008.

[47] P. Merkle, A. Smolic, K. Muller, and T. Wiegand, "Efficient prediction structures for multiview video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 11, pp. 1461–1473, 2007.

[48] K. Muller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 643–656, 2011.

[49] C. Fehn, "A 3D-TV system based on video plus depth information," in *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers*. IEEE, 2003, vol. 2, pp. 1529–1533.

[50] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Motion vector sharing and bitrate allocation for 3D video-plus-depth coding," *EURASIP Journal on Applied Signal Processing*, p. 3, 2009.

[51] K. Klimaszewski, K. Wegner, and M. Domanski, "Distortions of synthesized views caused by compression of views and depth maps," in *3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*. IEEE, 2009, pp. 1–4.

[52] W. Kumwilaisak and F. Hartung, "An intraframe error concealment: nonlinear pattern alignment and directional interpolation," in *International Conference on Image Processing, ICIP'04*. IEEE, 2004, vol. 2, pp. 825–828.

[53] J. Zhang, J. F. Arnold, and M. R. Frater, "A cell-loss concealment technique for MPEG-2 coded video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 4, pp. 659–665, 2000.

[54] S. Chen and H. Leung, "A temporal approach for improving intra-frame concealment performance in H.264/AVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 3, pp. 422–426, 2009.

[55] M. Yang, Y. Yang, and P. Cosman, "Depth-assisted error concealment for intra frame slices in 3D video," in *19th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2012, pp. 1281–1284.

[56] Y. Yang, M. Yang, and P. Cosman, "A hybrid error concealment for intra frame in stereoscopic video," in *Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*. IEEE, 2012, pp. 1104–1108.

[57] W.-M. Lam, A. R. Reibman, and B. Liu, "Recovery of lost or erroneously received motion vectors," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-93.* IEEE, 1993, vol. 5, pp. 417–420.

[58] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 966–976, 2000.

[59] D. Kontopodis T. Stockhammer and T. Wiegand, "Rate-distortion optimization for JVT/H.26L coding in packet loss environment," *Proc. Packet Video Workshop*, Apr. 2002.

[60] Y. Zhang, W. Gao, Y. Lu, Q. Huang, and D. Zhao, "Joint source-channel rate-distortion optimization for H.264 video coding over error-prone networks," *IEEE Transactions on Multimedia*, vol. 9, no. 3, pp. 445–454, 2007.

[61] J. Seo, D. Park, H. C. Wey, S. Lee, and K. Sohn, "Motion information sharing mode for depth video coding," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE, 2010, pp. 1–4.

[62] H. Oh and Y.-S. Ho, "H. 264-based depth map sequence coding using motion information of corresponding texture video," *Advances in Image and Video Technology*, pp. 898–907, 2006.

[63] C. T. Hewage, S. Worrall, S. Dogan, and A. M. Kondoz, "Frame concealment algorithm for stereoscopic video using motion vector sharing," in *IEEE International Conference on Multimedia and Expo*. IEEE, 2008, pp. 485–488.

[64] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Motion vector sharing and bitrate allocation for 3D video-plus-depth coding," *EURASIP Journal on Applied Signal Processing*, vol. 2009, pp. 3, 2009.

[65] D. De Silva, W. Fernando, and S. Worrall, "3D video communication scheme for error prone environments based on motion vector sharing," in *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*. IEEE, 2010, pp. 1–4.

[66] T. Wiegand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 2, pp. 182–190, 1996.

[67] S. Wenger, "Common conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing," *ITU-T SG16 Doc. VCEG-N79r1*, vol. 44, 2001.

[68] G. J. Han, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," 2012.

[69] J.-W. Suh and Y.-S. Ho, "Error concealment based on directional interpolation," *IEEE Transactions on Consumer Electronics*, vol. 43, no. 3, pp. 295–302, 1997.

[70] A. Raman and M. Babu, "A low complexity error concealment scheme for MPEG-4 coded video sequences," in *IEEE Symposium on Multimedia Communications and Signal Processing*. Citeseer, 2001.

[71] C. Liu, R. Ma, and Z. Zhang, "Error concealment for whole frame loss in HEVC," in *Advances on Digital Television and Wireless Multimedia Communications*, pp. 271–277. Springer, 2012.

[72] H. Sasai, S. Kondo, and S. Kadono, "Frame-rate up-conversion using reliable analysis of transmitted motion information," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'04)*. IEEE, 2004, vol. 5, pp. 257–260.

[73] A.-M. Huang and T. Nguyen, "Motion vector processing based on residual energy information for motion compensated frame interpolation," in *IEEE International Conference on Image Processing*. IEEE, 2006, pp. 2721–2724.

[74] A.-M. Huang and T. Nguyen, "A novel multi-stage motion vector processing method for motion compensated frame interpolation," in *IEEE International Conference on Image Processing, ICIP.* IEEE, 2007, vol. 5, pp. V–389.

[75] Reference software of HEVC (HM): https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.

[76] J. Nightingale, Q. Wang, and C. Grecos, "Hevstream: a framework for streaming and evaluation of high efficiency video coding (HEVC) content in loss-prone networks," *IEEE Transactions on Consumer Electronics*, vol. 58, no. 2, pp. 404–412, 2012.

[77] J. Leng, L. Sun, T. Ikenaga, and S. Sakaida, "Content based hierarchical fast coding unit decision algorithm for HEVC," in *Multimedia and Signal Processing (CMSP), 2011 International Conference on*. IEEE, 2011, vol. 1, pp. 56–59.

[78] H.-Y. Chen, "An efficient fast CU depth and PU mode decision algorithm for HEVC," 2011.

[79] ITU-T SG16 WP3/ ISO/IEC JTC1/SC29/WG11 JCTVC-H0072, "NAL Unit Loss Software," 2012.