UNIVERSITY OF CALIFORNIA, SAN DIEGO

Video packet loss visibility models and their application to packet prioritization

A Dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy

in

Electrical Engineering
(Signal and Image Processing)

by

Ting-Lan Lin

Committee in charge:

Professor Pamela C. Cosman, Chair
Professor Yoav Freund
Professor William S. Hodgkiss
Professor Truong Q. Nguyen
Professor Nuno Vasconcelos

2010

The Thesis of Ting-Lan Lin is approved and it is acceptable in quality and form for publication on microfilm and electronically:

Chair

University of California, San Diego

2010

DEDICATION

To my parents, my wife and my two inspiring daughters

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank the people who have helped me overcome the difficulties I encountered during my PhD study in UCSD in the past five years.

In the development of my academic profession, I thank my advisor, Prof. Pamela Cosman, for being the main source of inspiration. I always gain knowledge and find my weakness during our regular meetings. Every time when I proposed a new idea, she always responded with constructive feedback that was very helpful for me to develop my next step. She is also a great teacher in logical thinking and English writing. Every time when she reviewed my drafts that were going to be submitted for publication, I received useful criticism on my writing style from her. That improves my understanding and skill at technical writing. She also taught me how to give an organized and understandable presentation for different groups of people with different knowledge background. What I learned from her in my PhD study will be very useful for my future career.

I would like to thank another advisor of mine, Dr. Amy Reibman from AT&T Labs, for the remote instruction on finishing my first journal paper. I am grateful for her patience on constantly writing long and detailed emails to us in UCSD to explain her inspiring ideas. Her professional knowledge in video compression and networking truly helped me develop my research background.

I would like to thank my dissertation committee members, Prof. Truong Nguyen, Prof. William Hodgkiss, Prof. Nuno Vasconcelos and Prof. Yoav Freund, for their invaluable time and feedback in my PhD Qualification exam and Defense exam. The suggestions from you have strengthened the dissertation.

I am also thankful to my friends in the lab for positive interactions. I want to thank Athanasios Leontaris and Sandeep Kanumuri for introducing me to the world of video compression in the early stage of my research progress. I also want to thank Mayank Tiwari and Hobin Kim for providing the knowledge of their research details to help me broaden and expand my research area. I want to thank Jihyun Shin and Yueh-Lun Chang for collaboration on different subjective experiments. I want to thank visiting scholars/students from China for their friendship. Special thanks to Yuxia (Flora) Wang for cooperation on several conference papers.

I am thankful to all the friends in the Taiwanese Lutheran Church of San Diego. Although I am not a member of the organization, many sincere friends in the church weekly pray for the progress of my research and the wellness of my family. I really appreciate it.

I want to express my deepest gratitude to my family members for their unconditional support, whether it is from Taiwan or San Diego. I want to thank my wife, Sin-Mei, for giving me and taking great care of my two lovely daughters in San Diego. Her effort and sacrifice for the family are invaluable. She is the greatest support of the family and I thank her for the understanding of my constantly overtime working schedule. I want to thank my daughters, Mia and Kate, for providing surprising but pleasant break time during my continuous and exhausting research study. I am greatly thankful to my mother Shu-Ling Lin Cheng and my father Chia-Hao Lin for the encouragement and support for me to pursue higher education in prestigious schools. Your care from the other end of the world can only be stronger and warmer than ever. I also thank my sisters Ting-Hsuan and Ting-Chien for bringing happiness and fun to the family.

Chapters II and III of this dissertation, in part, are a partial reprint of the material as it appears in T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman and A. 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, March 2010. I was the primary author and the co-authors Prof. Cosman and Dr. Reibman directed and supervised the research which forms the basis for Chapters II and III. The co-authors Dr. Kanumuri and Dr. Poole also contributed to the ideas in this work. The co-author Y. Zhi helped the simulation process.

Chapter IV of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin and P. Cosman, "Optimal RCPC channel rate allocation in AWGN channel for perceptual video quality using integer programming", *QoMEX 2009, First International Workshop on Quality of Multimedia Experience*, July, 2009, and in T.-L. Lin and P. Cosman, "Perceptual Video Quality Optimization in AWGN Channel Using Low Complexity Channel Code Rate Allocation", *43rd Asilomar Conference on Signals, Systems and Computers*, 2009, and in T.-L. Lin and P. Cosman, "Efficient optimal RCPC code rate allocation with packet discarding for pre-encoded compressed video", *IEEE Signal Processing Letters*, vol. 17, issue 5, pp. 505-508, 2010. I was the primary author of these three papers. Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter IV.

Chapter V of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin and P. Cosman, "Network-based packet loss visibility model for SDTV and HDTV for H.264 videos", *International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2010.* I was the primary author. Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter V.

Chapter VI of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin, J. Shin and P. Cosman, "Packet dropping for widely varying bit reduction rates using a network-based packet loss visibility model", *IEEE Data Compression Conference (DCC), 2010.* I was the primary author. Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter VI. Co-author J. Shin also contributed to the ideas in this work.

Chapter VII 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 conference 2010.* I was the primary author. Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter VII. Co-author Y.-L.

Chang also contributed to the subjective experiment in this work.

VITA

1979	Born, Taoyuan, Taiwan
June 2001	B.S., Department of Electronic Engineering Chung Yuan Christian University, Chung-Li, Taiwan
July 2001–June 2003	Research & Teaching Assistant, Department of Electronic Engineering Chung Yuan Christian University, Chung-Li, Taiwan
June 2003	M.S., Department of Electronic Engineering Chung Yuan Christian University, Chung-Li, Taiwan
Dec 2003–Jan 2005	Special Engineering Officer (2nd Lieutenant) Communications Development Office of Ministry of National Defense, Taiwan
Fall 2006	Teaching Assistant, University of California, San Diego
Sept 2005–Dec 2010	Research Assistant, University of California, San Diego
Winter 2008 Summer 2008	Intern, Qualcomm Inc, San Diego, California
Dec 2010	Ph.D., Electrical Engineering (Signal and Image Processing) University of California, San Diego

PUBLICATIONS

Communication Theory and System

Journal Papers

Y.-H. Chang and T.-L. Lin, "Zero-forcing blind equalizer based on the transformed eigen decomposition", *IEE Electronics Letters*, Volume 40, Issue 24, pp. 1558-9, Nov.25, 2004 .

Conference Papers

Y.-H. Chang and T.-L. Lin, "Blind equalizers of multiple FIR channels based on the structure of channel coefficient matrix", *National Symposium on Telecommunication*, Taiwan, 2002.

Y.-H. Chang and T.-L. Lin, "Blind channel identification based on the noise-free correlation matrix", *International Symposium on Communication (ISCOM, in cooperation with IEEE Communications Society-Taipei Chapter)*, Taiwan, 2003.

Signal and Image Processing

Journal Papers

T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman and A. 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, March 2010.

T.-L. Lin and P. Cosman, "Efficient optimal RCPC code rate allocation with packet discarding for pre-encoded compressed video", *IEEE Signal Processing Letters*, vol. 17, issue 5, pp. 505-508, 2010.

Conference Papers

T.-L. Lin, P. Cosman and A. Reibman, "Perceptual impact of bursty versus isolated packet losses in H.264 compressed video", *IEEE International Conference on Image Processing*, ICIP, 2008.

T.-L. Lin, Y. Zhi, S. Kanumuri, P. Cosman and A. Reibman, "Perceptual quality based packet dropping for generalized video GOP structures", *International Conference on Acoustics, Speech, and Signal Processing*, ICASSP, 2009.

T.-L. Lin and P. Cosman, "Optimal RCPC channel rate allocation in AWGN channel for perceptual video quality using integer programming", *QoMEX 2009, First International Workshop on Quality of Multimedia Experience* , July, 2009.

T.-L. Lin and P. Cosman, "Perceptual video quality optimization in AWGN channel using low complexity channel code rate allocation", *43rd Asilomar Conference on Signals, Systems and Computers*, 2009.

T.-L. Lin and P. Cosman, "Network-based packet loss visibility model for SDTV and HDTV for H.264 videos", *International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2010.*

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

Y. Wang, T.-L. Lin, and P.C. Cosman, "Network-based model for video packet importance considering both compression artifacts and packet losses", *IEEE Globecom 2010.*

Y. Wang, T.-L. Lin and P. C. Cosman, "Packet dropping for H.264 videos considering both coding and packet-loss artifacts", *IEEE Packet Video Conference 2010.*

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 Conference 2010.*

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 Conference 2010.*

FIELDS OF STUDY

Major Field: Electrical Engineering

Studies in Signal and Image Processing.
Professor Pamela C. Cosman.

Studies in Communication Theory and Systems.
Professor Yuh-Huu Chang, Chung Yuan Christian University, Chung-Li, Taiwan.

ABSTRACT OF THE DISSERTATION

Video packet loss visibility models and their application to packet prioritization

by

Ting-Lan Lin
Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)
University of California, San Diego, 2010
Professor Pamela C. Cosman, Chair

In video transmission, packets can be lost for many reasons. Traditionally the impact of packet losses is measured by mean squared error induced by the loss in the pixel domain. However, mean squared error does not correlate with human perception well. In this dissertation, we aim to provide predictions of how human observers respond to different video packet losses. Based on their estimated visual importance, we can insert a prioritization bit for each video packet before sending it over a lossy network, or perform unequal channel protection on packets before transmission over a wireless channel. The models are developed from data collected from subjective tests. The models predict the packet loss visibility, that is, the probability of a given packet producing a glitch that will be observed by the end user if it is lost. We discuss the development and the application of encoder-based packet loss visibility models and network-based packet loss visibility models.

We discuss an encoder-based packet loss visibility model using three subjective experiment data sets that span various encoding standards (H.264 and MPEG-2), group-of-picture structures, and decoder error concealment choices. The factors of scene cuts, camera motion, and reference distance are highly significant to the packet loss visibility. The encoder-based packet loss visibility model exploits factors in the pixel domain as well as reference frame information.

The first application of the encoder-based packet loss visibility model is packet prioritization for a video stream. When a network gets congested at an intermediate router, the router is able to decide which packets to drop such that visual quality of the video is minimally impacted. Experiments are done to compare our perceptual-quality-based packet prioritization approach with existing Drop-Tail and cumulative-MSE-based prioritization methods. The result shows that our prioritization method produces videos of higher perceptual quality for different network conditions and group-of-picture structures.

The second application of the encoder-based packet loss visibility model is unequal error protection. For an AWGN channel, we aim to minimize the end-to-end video quality degradation using rate-compatible punctured convolutional codes for a given channel rate budget. We solve the integer programming problem by the Branch and Bound method, K-means clustering, and the subgradient method. We also exploit the advantage of not sending or not coding packets of lower importance. The algorithm is compared to an existing method.

In order to reduce the computational complexity of the encoder-based model so that a model can be implemented at the router, we aim to develop a network-based model that uses only information within one packet to predict the importance of that packet, requiring no frame-level reconstruction nor any information on the reference frame. We conduct subjective experiments for SDTV and HDTV resolutions on visual quality following packet loss. We design the model for SDTV and HDTV resolutions, and discuss the differences in the important factors between SDTV and HDTV models.

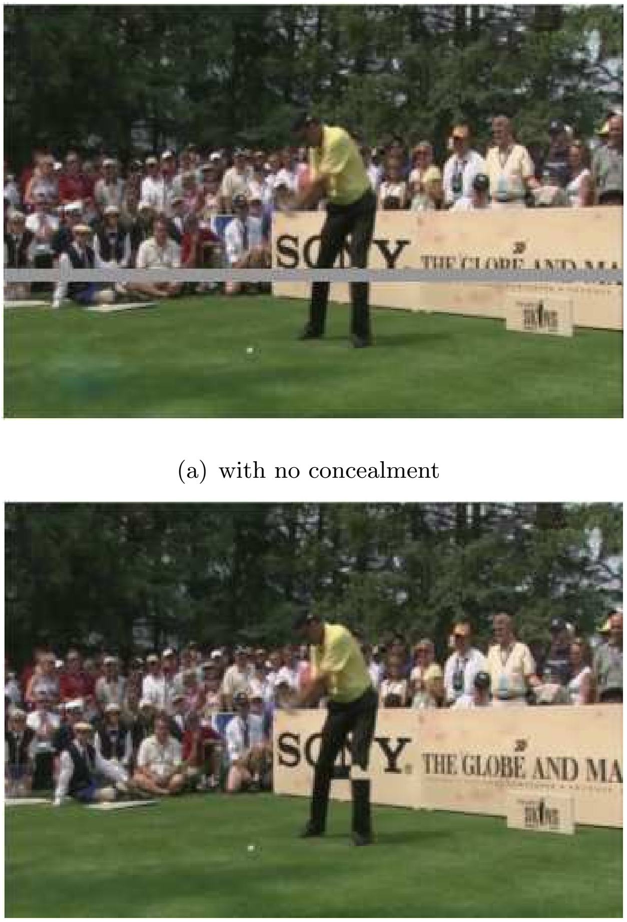
We then use the model to measure the visual importance of incoming packets to the router. During network congestion, we drop the least visible frames and/or the least visible packets until the required bit reduction rate is achieved. Our algorithm performs better than dropping B packets/frames.

The way we estimated the frame importance is based on the summation of the visibility of all slices in a frame, which is an indirect approach. Therefore, we conduct subjective experiments and collect responses from human observers directly on whole frame losses. We develop a model which can predict the visibility of whole frame losses for B frames. This model could be useful for designing an intelligent frame dropping approach for use at a router during congestion.

I

Introduction

Video transmission over wired and wireless networks is very popular due to the high demand of multimedia and widespread availability of personal web surfing devices. However, as compressed videos are transmitted over a network, they can suffer losses which affect overall video quality for several reasons. For example, packets may be dropped in a router to relieve network congestion, or bit errors occur in the packet. To deal with these situations, we can perform packet prioritization and unequal error protection over packets so that the algorithms can protect packets of higher importance. Therefore, it is very important to accurately predict how human observers respond to different packet losses, which is the main focus of this dissertation. In Figure I.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 I.1(b) the compressed and reconstructed frame is shown where the loss has been concealed by holding over the corresponding blocks from the previous frame. The glitch in this case is visible. Figures I.2(a) and (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. In this work, we measure the importance of a packet by performing subjective experiments and using the data to develop models to predict the probability that a certain packet will produce an observable glitch if it is lost. We call this probability

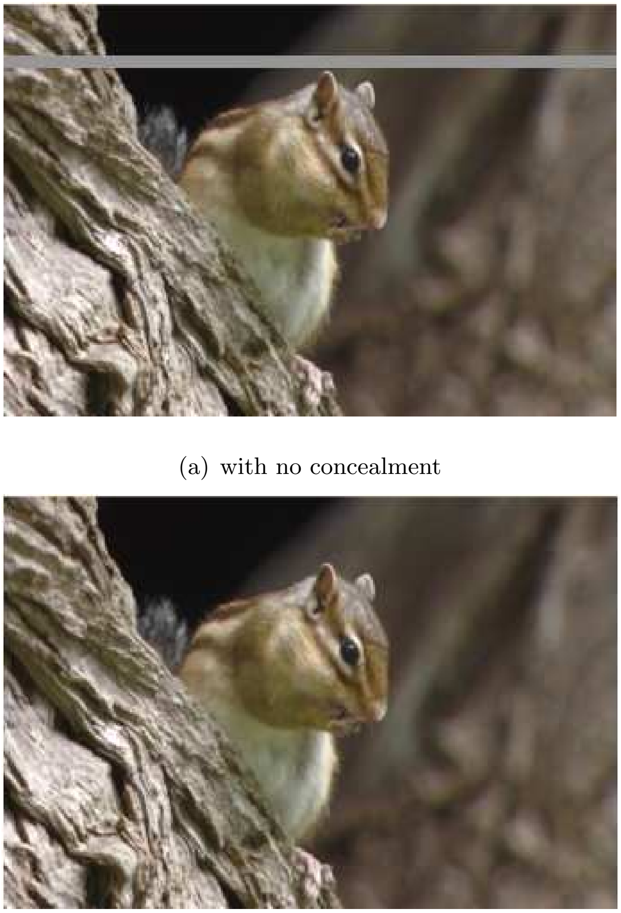


(a) with no concealment

(b) after concealment

Figure I.1: A frame with packet loss (a) with no concealment (b) after concealment. The loss is visible.

packet loss visibility. Based on this measurement, we develop applications such as intelligent dropping and unequal error protection. Our work covers two categories: *encoder-based* packet loss visibility and *network-based* packet loss visibility. In the following, we introduce each of them and their applications, along with relevant literature.



(a) with no concealment

(b) after concealment

Figure I.2: A frame with packet loss (a) with no concealment (b) after concealment. The loss is less visible.

I.A Encoder-based packet loss visibility

To ensure a satisfactory viewing experience for the end users, it will be beneficial for network providers or video transmitters to have an accurate video quality monitoring system to help evaluate the quality of video reception. A good video quality monitoring system can help video senders or network service providers decide/optimize the transmission settings that result in efficient usage of network resources (for example, bandwidth) for video services with cost-based quality.

Video quality measurement can be categorized into three different types based on the accessibility of information about the original (reference) video. Full-reference (FR) methods evaluate the video quality with access to the original video, providing the most precise measurements on the video quality difference. Reduced-reference (RR) metrics extract partial information about the original video at the sender and are sent reliably to the receiver to estimate the video quality. No-reference (NR) methods only use information available in the bitstream (NR-B) or the decoded pixels (NR-P) without reference video information. These methods are illustrated in Figure I.3.

One of the most widely used FR metrics is MSE (Mean Squared Error) of pixel values between original and evaluated videos. The Structural SIMilarity index for images (SSIM) [1] and for videos (VSSIM) [2] also requires the original content to calculate statistical structure information. Another FR metric is Continuous Video Quality Evaluation (CVQE) [3], which models the temporally continuous quality scores of human observers. Video Quality Metric (VQM) [4], a FR metric developed by the National Telecommunication and Information Administration, has been shown to be better correlated with human perception than two competing metrics, DVQ (Digital Video Quality) and VSSIM, as shown in [5]. A RR method developed in [6] sends a low-bandwidth descriptor which approaches the performance of VQM. In [7], harmonic analysis on filtered images is done to provide a RR metric, which shows good correlation with subjective data in the VQEG database. A NR method proposed in [8] evaluates blurring artifacts using edges and adjacent regions in the lossy image. In [9], PSNR of a lossy video is estimated by a NR metric using only received coded transform coefficients.

Packet losses in the network can significantly damage video quality during transmission. They occur for different reasons. An intermediate router can drop packets because the incoming data rate is so high that the buffer overflows. With IPTV, a subscriber may want to watch a video in high resolution, but his access bandwidth is less than required. In this situation, a router should drop enough data to meet the access capabilities of the subscriber. The packet dropping policy in the router should be intelligent to minimize the video quality damage observed

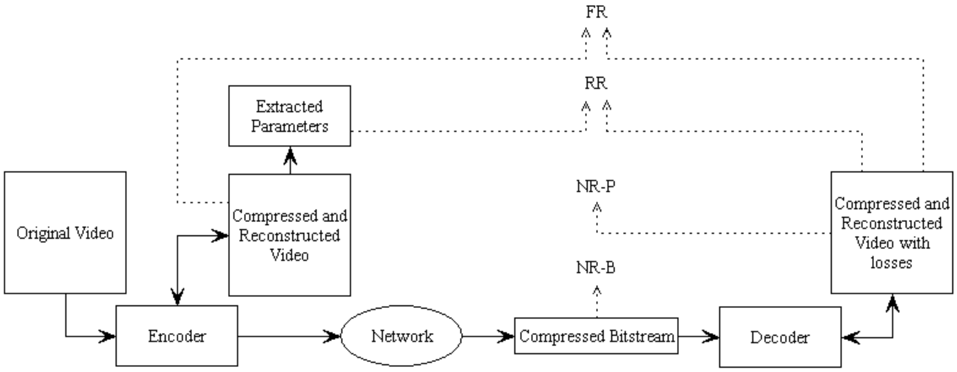


Figure I.3: FR, RR, NR-P and NR-B methods

by the end user. The packet dropping rates required at the router can vary by a large amount. Therefore, considerable research has been conducted to understand the relationship between packet losses and visual quality degradation. Although PSNR (Peak Signal to Noise Ratio) and MSE do not always reflect perceptual quality well [10, 11], they are commonly used to measure video quality. The relation between PSNR and perceptual quality scores is considered in [12]. It finds that packet losses are visible when the PSNR drop is greater than a threshold, and the distance between dropped packets is crucial to perceptual quality. The prediction of objective distortion by MSE is discussed in [13]. Average performance across an entire video sequence is the focus in [14], which uses MSE to assess quality for different compression standards and different concealment techniques; a specific model is used for each compression standard and concealment technique. Three different NR metrics in [15] are developed to estimate the MSE caused by a packet loss.

Much of the effort to understand the visual impact of packet losses [16, 17, 18, 19] has focused on modeling the average quality of videos as a function of average packet loss rate (PLR). Video conferencing was studied in [16] using the average consumer judgments on the relative importance of bandwidth, latency and packet loss. A random-neural-network model was developed in [18] to assess quality given different bandwidths, frame-rates, packet loss rates, and I-block refresh rates.

In [20, 21], NR metrics of low complexity were developed using the length and strength of packet loss impairment from each decoded image.

However, PLR can provide wrong interpretations on video quality since packet losses are perceptually not equal. The visual impact of a packet loss is a combined effect of various factors such as the location of the packet loss in the video, the content of the video at that location, and whether there are other packet losses in its vicinity. Therefore subjective experiments are important to construct/verify the video quality metrics related to packet losses. Hughes et al. [19] discovered that many different realizations of both packet loss and video content are necessary to reduce the variability of viewer responses. Also the "forgiveness effect" causes viewers to rank a long video based on more recently viewed information. In [22], using computational metrics from a no-reference model as well as a subjective test, it was found that simple quality metrics (such as blockiness, blurriness and jerkiness) do not predict quality impairments (caused by packet losses or compression) very well. In [23], an NR metric was used to calculate the temporal fluidity impairments resulting from packet losses. This is a good predictor for the perceptual scores due to motion discontinuity under several image dropping conditions.

Instead of studying how packet losses affect the overall perceptual video quality, or how packet losses relate to MSE, our goal is to develop a robust predictor for packet loss visibility for *each individual packet* based on the information of its encoded content and other factors. This will serve as a useful tool for various purposes. The first one is *packet prioritization.* One can assign low priority to packets that cause low loss visibility. When the buffer in a network node is congested, it can opt to discard the low-priority packets and hence minimize the degradation to perceived video quality for the end user. For *unequal error protection*, one can give more parity bits and hence more protection to packets with higher visual importance, so that if those packets are corrupted during transmission, they are more likely to be corrected by the channel decoder.

The closest previous work to our research is [24, 25, 26, 27, 28, 29]. In [24, 25, 26, 27], as in our research, subjective experiments were performed to collect and analyze data from human observers. Isolated packet losses were introduced in the video and viewers were asked to observe the videos and respond to the visible glitches that they notice. Based on the data and the coding parameters associated with each packet, regression models for prediction were built. Papers [24, 25, 26] used MPEG-2 and [27] used H.264. In [28], information from the neighboring slices of a packet, both spatially and temporally, is further considered. In [29], more complicated factors such as scene cut information and camera motion are included. The model uses data from multiple codecs, video resolutions, GOP (Group Of Pictures) structures, and error concealment methods. The best model that achieves the least prediction error is presented in Chapter II. This model, which is developed in large part by Dr. Sandeep Kanumuri and Dr. Amy Reibman, is used in Chapters III and IV of this dissertation in packet prioritization and unequal error protection applications.

I.A.1 Packet prioritization

One application for the visibility model is packet prioritization. A network can sometimes become congested due to sudden traffic. To relieve the congestion, the router needs to drop packets. Therefore it is important that the router can identify important packets and make sure they are not dropped. In Chapter III we use the visibility model described in Chapter II to perform packet prioritization. Using the packet priorities, an intermediate router can intelligently drop low-priority packets.

Among existing approaches on packet classification, the work by De Martin *et al.* assigns a packet high/low priority based on the cumulative MSE due to the packet loss [30, 31], network status and end-to-end QOS constraint [32]. Also, the Rate-Distortion Hint-Track method was proposed for packet scheduling [33, 34] and packet dropping [35, 36]. Especially in [35, 36], an intermediate router with an optimization algorithm drops packets in a congested network from different streams to minimize the sum of the cumulative MSE, where the sum of the outgoing rates is constrained to be less than the bandwidth of the outgoing link. A similar idea on a rate-distortion optimized dropping policy was proposed earlier in [37] using a rate vector and a distortion matrix, and employing a different optimization philosophy. A detailed discussion of the Hint-Track (HT) and Distortion Matrix (DM) methods is in [38]. The most significant difference between our approach and the above-mentioned methods is that we do not use MSE (or PSNR) as a quality metric to develop our method; the model is built from subjective experiments. We compare our visibility-based packet prioritization strategy with the Cumulative-MSE-based method and the widely-used Drop-Tail policy using the NS-2 Network Simulator [39]. The comparisons are made using VQM. This work is presented in Chapter III.

I.A.2 Forward error protection

A video transmitter can also protect important packets by channel coding. Joint source and channel coding that trades between source video quality and error resilience of the transmission in a lossy network is a well-studied area. The work in [40] studied a combined source-channel coding problem for transmission in an AWGN channel using Rate-Compatible Punctured Convolutional (RCPC) codes. A universal operational distortion-rate characteristic is used for the optimization problem, and the performance of the algorithm approaches the information-theoretic bound. The rate-distortion optimization among source coding, channel coding and error concealment is jointly considered in [41]. A selective packet retransmission mechanism is integrated into the algorithm. Unequal error protection for joint source and channel coding is considered in [42] using a source and channel distortion function to find the best source and channel rate allocation, where the channel codes used are RCPC codes. The method in [43] optimally decides both slicing based on the estimated incurred distortion, and optimal forward error correction (FEC) rate for each packet. In [44], H.264 Flexible Macroblock

Ordering (FMO) is used to group macroblocks of similar estimated distortion into a slice, with different levels of Reed-Solomon (RS) coding over slices. This method is extended in [45] with Converged Motion Estimation, which performs motion estimation for the current frame using mostly the highly-protected MBs (Macroblocks) in the previous frame as reference.

Unequal error protection (UEP) tailors the error-handling measures to different components of the video. For example, in [46], frames closer to the end of a GOP have less error protection. Two different RS codes are assigned to video data of high/low priorities in [47]. In [48], channel protection bits are assigned unequally among frames in a GOP using the Genetic Algorithm.

UEP can be jointly used with intra updating which stops error propagation. This option requires more source and channel bits to intra-code the slice, therefore there is a tradeoff among video quality, error correctability, and the ability to stop error propagation in case of uncorrectable errors [49, 50, 51]. UEP for progressively coded images/videos is also extensively studied [52, 53].

Most of the work on UEP discussed above involves either progressive or scalable coding, or a change to the source encoder. In our work, we consider non-scalable video streams pre-encoded and stored; the problem is choosing optimal packet protection for the channel conditions at the time of transmission. Also, traditionally, video quality degradation is measured with MSE. To improve the performance of channel rate allocation in terms of human visual perception, we use the encoder-based model form Chapter II to estimate the visual importance of each packet. Based on this metric, we aim to optimally allocate RCPC codes to minimize the visual quality degradation when transmitting video over an AWGN channel, given a total rate budget. We solve the optimization problem first by the Branch and Bound method (BnB) [54]. However, BnB has worst case exponential complexity [55]. We reduce the complexity of the algorithm by preprocessing the packet information using k-means clustering [56]. We further devise an algorithm where the optimal FEC code rate allocation search is done efficiently in the dual domain by the method of subgradients, and we also include the options of not coding a packet, or not sending it at all. Those modifications improve the performance significantly in terms of VQM measurement. We also compare with an existing approach [57] that uses the notion of not coding and not sending a packet. This research is presented in Chapter IV.

I.B Network-based packet loss visibility

In the application of packet dropping described above, the incoming packets to the router must have a prioritization bit embedded for the router to perform intelligent dropping. To cover the common situation where the incoming packets do not have a prioritization bit, we focus on developing a *network-based model* where the packet importance is estimated at the router. In this case, the complexity must be limited, and reference frames are not necessarily available because packets may be out of order or because there are multiple streams and the network node cannot afford to decode and reconstruct them. In a network-based model therefore, we refrain from using factors such as MSE, and scene cut information that requires pixel domain information and reference frame information. To this end, we perform a subjective experiment, for SDTV and HDTV resolutions. We build network-based models for each resolution and compare their prediction accuracies against those of the encoder-based models. We also discuss some differences of the human responses between SDTV and HDTV videos. Some visual differences between SDTV and HDTV have been discussed in previous literature [58, 59, 60]. The development and the discussion of these models are covered in Chapter V.

I.B.1 Intelligent packet discarding

Using this network-based model, we can produce visual scores on the fly at a router with limited information and limited computational power, and perform packet dropping. Prior research [61, 62] considered No-Reference network monitoring, which can compute estimated video quality for a given packet loss pattern. However, they give an overall quality score for the sequence and do not tell us how to best drop packets to minimize the video quality degradation during congestion. In this dissertation, using the visual scores computed by the network-based model, we drop packets on a packet basis and on a frame basis. The dropping method on a frame basis is visually better than the method on a packet basis since spatial misalignment artifacts can be more distracting than temporal frame copy [63]. Our method performs better than one used by industry which drops B packets, for different levels of packet loss rate, where the video quality is evaluated by VQM. Chapter VI covers the discussion of dropping methods using our network-based model.

I.C Network-based whole frame loss visibility

Which whole frame to be dropped in Chapter VI was estimated by the visibility model for single-slice packets from Chapter V. That is, the visibility score for the frame was taken to be simply 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 involving slice loss data. Therefore, to obtain more meaningful scores for frame losses, we conduct a subjective experiment to concentrate on the subjective results for whole frame loss, and build a direct model for whole frame loss. Two common concealment methods for whole frame losses are frame copy and temporal frame interpolation. In this experiment, we simulate frame copy error concealment by the JM standard decoder [64], and frame interpolation by FFMPEG [65]; these two decoders are popular in research and industry. We analyze the experimental data, and model the whole frame packet loss visibility based on information associated with the lost frames. In the literature, the perceptual quality of frame losses is discussed in [66, 67]. However, the video quality is computed in the pixel domain and requires the original video. Also, their model aims to evaluate the overall quality of a lossy video, and does not indicate the visual importance of a specific frame. The discussion of our experiment is in Chapter VII.

I.D Introduction to generalized linear models

Our goal in several sections of this dissertation is to develop a model that predicts the probability of a glitch from a lost packet being visible to viewers. We first conduct subjective experiments. Viewers watch videos with isolated packet losses, and they are asked to press the space bar when they notice a glitch. In some experiments, each loss was observed by ten people, while in other experiments each loss was seen by twelve people. The ground truth packet loss visibility is the number of people who observe the packet loss divided by 10 (or 12) people. In our experiment and data analysis, 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 independent and identically distributed with probability p for seeing a particular packet loss. This leads us to the binomial distribution for modeling the packet loss visibility.

A *Generalized Linear Model* (GLM) is an extension of classical linear models [68, 69]. The probability of visibility is modeled using logistic regression, a type of GLM which is a natural model to predict the parameter p of a binomial distribution [68]. 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

$$g(p_i) = \gamma + \sum_{j=1}^{P} x_{ij}\beta_j \tag{I.1}$$

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 γ are usually unknown and need to be estimated from the data. 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

$$g(p) = log(\frac{p}{1-p}). \tag{I.2}$$

Given N observations, one can fit models using up to N parameters. The simplest model (Null model) has only one parameter: the constant γ. At the other extreme, it is possible to have a model (Full model) with as many factors as there are observations. The parameters and coefficients of the GLM are estimated such that the resulting model has the least deviance (the deviance is a generalization of the residual sum of squares). This method is used in Chapter II.

The method treats data points equally, no matter how far they are from the regression line. However, outliers may distort the results. To give unequal treatment to data points to suppress outliers, we minimize the M-estimator [70]; data points farther from the regression line have smaller weights, and contribute less to the final modeling result. We chose the “Fair” function as the M-estimator function, shown in Figure I.4. The M-estimator is computed as the sum of the weighted residual squares, where the weight of each data point is computed by the residuals in the previous iteration. The M-estimator function is chosen to avoid the weights of the curve going close to zero at the two ends, because we do not want to have a final model that has least M-estimator just because most of the data points are at the two ends. The model developing procedure uses 4-fold cross validation to prevent the model overfitting the data, so an average M-estimator is produced for a set of factors. The factor which most reduces the average M-estimator goes next into the model. This procedure repeats until there is no improvement in the average M-estimator by including an additional factor. This method is used in Chapters V and VII.

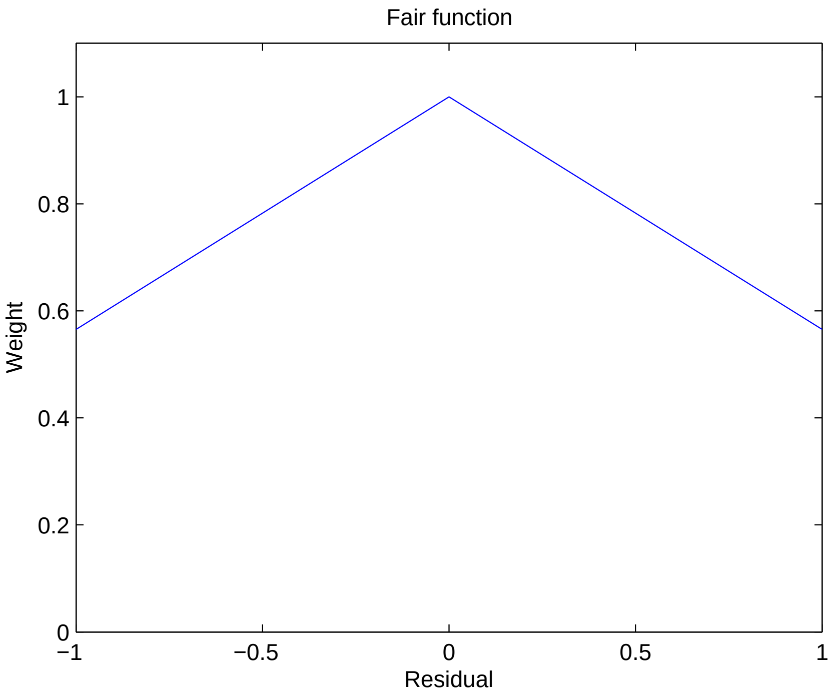


Figure I.4: The Fair function versus the residual.

I.E Thesis outline

In Chapter II, we provide detailed background on an encoder-based packet loss visibility model that was largely developed by Dr. Sandeep Kanumuri and Dr. Amy Reibman. The description of the experiment settings for the three different subjective tests and the variety of settings used for video encoding are provided. The measurements of packet loss are explained in terms of required information about the video, computational complexity and factor attributes. The GLM model building strategy using all the different data sets and the incorporation of significant factors are discussed. Applications of this visibility model form the research in Chapters III and IV.

In Chapter III, we discuss the application of the encoder-based packet loss visibility model to packet prioritization. We present the experiment results comparing the visibility-based prioritization method with others.

In Chapter IV, we discuss the application of the encoder-based packet loss visibility model to unequal error protection. We formulate the RCPC rate allocation problem as an integer programming problem. Several different algorithms to find the solution are discussed. We discuss the simulation results of our algorithms, and compare our algorithm with one in the literature.

In Chapter V, we develop a network-based packet loss visibility model. The subjective tests are described. We discuss self-contained factors that relate to packet loss visibility, and the models based on these factors.

In Chapter VI, we discuss the application of the network-based packet loss visibility model to packet dropping. We propose a multiple packet loss algorithm using measurements obtained by the network-based packet loss visibility model. Simulation results are discussed.

In Chapter VII, we develop a network-based packet loss visibility model for whole frame loss. The setup of the subjective experiment is introduced. We cover the analysis of data, the whole frame loss modeling process and feature selection.

In the Conclusions section, we summarize the contributions of this dissertation, and discuss potential future work. We note that partial conclusions are also given at the end of each individual chapter.

II

Background: Encoder-based model building

In this chapter, we provide a background introduction on works related to the development of the packet loss visibility model.

The research in [24] studied the problem of predicting the visibility of individual packet losses in MPEG-2 bitstreams. Packet losses were introduced in MPEG-2 bitstreams and concealed using zero-motion error concealment (ZMEC). Viewers were asked to observe the videos and respond to the visible glitches that they notice. Using the subjective test results and a set of factors that were extracted from the videos, the Classification and Regression Trees (CART) algorithm [71] was applied to classify the losses as visible or invisible. This work was extended in [25] and [26] to model the probability of packet loss visibility using a generalized linear model (GLM) [68]. The visibility for H.264 packets was discussed in [27]. Visibility models specifically for individual and multiple packet losses based on RR factors were derived.

In all these studies, the main factors are based on encoding information within a slice (packet), such as motion vectors, residuals, and number of inter partitions. The work in [28] focused more on exploring features of the video frames in the pixel domain: encoded signal, decoded signal, and the error between them. Those factors are not only considered in the scope of a slice, but also for its neighboring slices, both spatially and temporally. For example, the work considered both the temporal and spatial edges induced by a packet loss, and also the error duration. In addition, [28] obtained a generic model that predicts the visibility of packet loss for two compression standards and three decoder concealment techniques without prior specification of either the standard or the concealment technique. The work in [29] considered factors related to the proximity of a scene cut and camera motion, and found their effectiveness to predict the visibility of packet loss. The Patient Rule Induction Method (PRIM) [72] was used to understand when the packet loss will be very visible and very invisible.

In this chapter, we discuss a new packet-loss visibility model based on a more general strategy for factor inclusion than in [28, 29]. There are many differences between this work and previous work [26], which used data from a single codec (MPEG-2), video resolution (720×480), GOP structure (IBBP), and error concealment method (ZMEC). This chapter uses data from multiple codecs, video resolutions, GOP structures, and error concealment methods. Also, the visibility model in [26] uses features which are specific to that GOP structure(IBBP). In this chapter, the GOP-specific variables are avoided. So the current model allows a far more generalized use, due to both the data used to build the model, and the choice of factors for prediction.

This chapter is organized as follows: Section II.A describes the experiment settings for the three different subjective tests and the variety of settings used for video encoding. Section II.B discusses constraints on factor extraction. Section II.C introduces the attributes of packet loss that can be extracted from the encoded signal, the decoded signal, and the error between them, to predict packet loss visibility. The measurements of packet loss are explained in terms of required information about the video, computational complexity and factor attributes. In Section II.D, we illustrate the GLM model building strategy using all the different data sets and incorporate significant factors.

II.A Subjective datasets

The major purpose of this work is to develop a generalized and robust visibility model for packet loss impairments. Therefore, the work considers the results of three prior subjective experiments [26, 27, 73] in which the video clips are generated by using various codecs and settings as summarized in Table II.1. The data sets used are the same as in [29].

Tests 1 and 2 use videos compressed by MPEG-2 at spatial resolution 720×480 with an adaptive GOP structure in which an I-frame is inserted at each scene cut. In these videos, there are usually two B-frames between each reference frame, and the typical GOP length is 13 frames. However, each GOP ends with a P frame and there are no B-frames between the final P-frame of one GOP and the first I-frame of the next GOP. Test 3 uses videos encoded by H.264/AVC extended profile (JM 9.1) at spatial resolution 352×240 with a fixed IBPBPB-type GOP structure of 20 frames. The encoder in this case uses each I-frame of the current GOP as a long-term reference frame. For P frames, a long-term reference frame and a short-term reference frame (previously-coded P frame) are used for motion compensation. B frames use the future P frame and either the long-term or short-term reference frame for bidirectional prediction. Test 3 does not enable the Flexible Macroblock Ordering (FMO) functionality in H.264. An important application of the desired visibility model is for high-quality video transmission over mostly reliable networks, where there are few, if any, visible compression artifacts and only isolated packet-loss events. Therefore, the encoding rates for all videos in the three tests were set such that there are no obvious encoding artifacts. This allows us to concentrate on impairments induced by packet loss. H.264 videos have one slice (a row of macroblocks) per Network Adaptation Layer Unit (NALU) by default, and each packet loss is equivalent to the loss of one slice. For MPEG-2 videos, the generic packet sizes are explored, by recognizing that a large variety of packet sizes can be accommodated by considering the loss of one slice, two slices (where a loss affects a slice header) or a full frame (where a loss may affect a picture header).

The main difference among the decoders is the concealment strategy, which is the most important factor influencing the initial error induced by a packet loss. Test 1 uses a default error concealment typical of a software decoder. The concealment uses the 2nd previous frame (previous to previous) in display order. This is because of the way the display buffers are updated. There are two display buffers - one is active for display while the other buffer is getting overwritten with the new frame getting decoded. When a slice is lost, the memory area corresponding to that slice will not be updated and so it will be an automatic concealment from the 2nd previous frame. Test 2 uses zero-motion error concealment (ZMEC), in which a lost macroblock is concealed using the macroblock in the same spatial location from the closest prior reference frame in display order. Test 3 uses Motion-Compensated Error Concealment (MCEC) [27], which incurs a lower initial error compared to ZMEC [24, 25, 26]. The MCEC algorithm estimates the motion vector and the reference frame for the lost macroblock and conceals it with the macroblock predicted using the estimated motion vector. Motion compensation in H.264/AVC can occur at different levels from the macroblock level to the smallest block level (4×4 pixel block). Accordingly, each macroblock can have a different number of motion vectors ranging from 1 to 16. These motion vectors can reference different reference frames because of multiple frame prediction. A set of motion vectors is formed from motion vectors of blocks around the lost macroblock. The frame that is referenced the most number of times in the set among all the reference frames is selected for concealment. The estimated motion vector is the median of all the motion vectors in the set that refer to this selected frame. The improved performance of MCEC can be seen from Table II.1. In Test 3, both the number of viewers observing each packet loss, and the initial MSE (IMSE), are reduced compared to Tests 1 and 2. Note that one common feature for the error-handling strategies of all the three decoders is that the video decoder only

Table II.1: Summary of subjective tests' parameters and their datasets

	Test 1 [73]		Test 2 [26]	Test 3 [27]
Spatial resolution	720x480		720x480	352x240
Frame rate (fps)	30	24	30	30
Duration of video in test (minutes)	7.3	8.9	72	36
Compression standard	MPEG-2		MPEG-2	H.264
GOP structure	I-B-B-P-		I-B-B-P-	I-B-P-
I-frame insertion	scene adaptive		scene adaptive	fixed
GOP length	≤ 13	≤ 15	≤ 13	20
concealment	default		ZMEC	MCEC
Losses	108	107	1080	2160
Losses in B-frames	14%		14%	50%
Full-frame losses	20%		30%	0%
Mean num. viewers who saw each loss	4.56	5.13	3.11	1.32
Null Pred. error	0.14599		0.12236	0.041571
Initial Mean Sq. Error (IMSE)	5.245		3.919	1.708

processes slices that are completely received.

The videos used in each test are highly varied in motion and spatial texture. They contain a wide variety of scenes with different types of camera motion (panning, zooming) and object motion. The high motion scenes include bike racing, bull fighting, dancing and flowing water. The low motion scenes include a slow camera pan of geographical maps, historic buildings and structures. The videos also have scenes with varying spatial content such as a bird's eye view of a city, a crowded market, portraits, sky and still water. The signal attributes of per-frame mean, variance, mean motion-vector length, and residual energy after motion compensation are all statistically identical across the three tests. The video content in Test 3 is identical to half the video content in Test 2, while the content in Test 1 is distinct and includes some content from film encoded at 24 fps.

The purpose of these subjective experiments is to obtain the ground truth on the visibility of packet losses. In each of these three tests, the viewers' task is to indicate when they saw an artifact, where an artifact is defined simply as a glitch or abnormality. All the subjective tests were single stimulus tests, which means that the viewers were only shown the videos with packet losses and not the original videos. A single stimulus test mimics the perceptual response of a viewer who does not have access to the original video, which is a natural setting for most applications. For each of the three tests, exactly one packet loss occurs in the first 3 seconds of every 4-second time slot, and the last second in the slot has no losses. This isolates the visual effect of one packet loss from another, and provides the viewer time to respond to the current loss before the next loss occurs. This was not intended to be a realistic simulation of a real network, rather, it was intended to provide information on the visibility of individual packet losses. However, the experimental section shows that the model is robust to various packet loss rates and to losses which may not be isolated. The distributions of the losses in the three tests are different. In Test 1, roughly 1/7th of all losses were forced to be in B-frames, 1/7th in I-frames, and 5/7th in P-frames, and roughly 20% of losses caused an entire frame to be lost. In Test 2, there is a similar ratio of losses in I/B/P frames, and roughly 30% of losses cause an entire frame to be lost. In Test 3, roughly half of the losses are in B-frames and about 5% of losses are in I-frames.

During the subjective test in all three tests, each packet loss was evaluated by 12 viewers. No more than one viewer for each packet loss was an expert viewer. A 1-minute pilot training video was shown to viewers, before the actual test, to help them understand the task and attain a basic level of expertise. Viewers were told that they will watch videos which are affected by packet losses. Whenever they see a visible artifact or a glitch, they should respond by pressing the space bar. They were asked to keep their finger on the space bar to minimize response time and ensure that this task did not take their attention away from the monitor. All tests were conducted in a well-lit office environment. Viewers were positioned approximately 6 picture heights from the CRT display. Based on comments from viewers after the tests, the full-color full-motion video was sufficiently compelling that they were immersed in the viewing process rather than searching for every artifact.

The output of the subjective test was a set of files containing the times that the viewer pressed the space bar relative to the start of the video. Once gathered, the data was processed as in [26] to obtain viewers' Boolean responses corresponding to whether they saw a loss or not. The ground-truth packet visibility was calculated as the number of viewers who saw the loss divided by 12.

II.B Constraints on factor extraction

The goal of the encoder-based packet loss visibility model is to describe the impact of losing this specific packet during transmission. Factors needed by the visibility model can either be extracted from the complete loss-free bitstream on the fly at the server when needed for transmission, or pre-computed and stored with the specific packet in the server. Any factors that depend on the uncompressed video must be computed at the encoder and sent to the server on a reliable channel along with the compressed video. However, factors that depend on the compressed video can be computed either at the server or at the encoder; the choice of where is up to the system constraints. However, in this work, to minimize the bandwidth between encoder and server that is required for these RR factors, only those based on the uncompressed video will be computed at the encoder. However, since the primary functions of the server are streaming and traffic shaping, factors computed here should not require excessive computation. In particular, any factors related to the propagation or accumulation of errors due to packet loss are not suitable to be computed here.

It is necessary to assume some knowledge of what concealment strategy is implemented by the actual decoder. For example, if using motion-compensated error concealment, the estimated motion depends on the motion of the neighboring received packets.

Section II.C describes the factors chosen to predict the visibility of packet loss, and indicates for each factor both whether it must be computed at the encoder or could be computed at the server.

II.C Attributes of packet-loss impairments

To create a versatile model for packet loss visibility, it is crucial to understand the types of impairments induced by a packet loss, and whether these impairments depend on (a) the codec and its parameters, (b) the packetization strategy, (c) the decoder error concealment and (d) the video content. In this section, these issues are explored by describing attributes that affect the visibility of packet loss impairments and the associated measurements. The following are defined to facilitate the discussion:

1. $f(t)$: the original signal of the uncompressed video frame at time t,
2. $\hat{f}(t)$: the compressed signal,
3. $\tilde{f}(t)$: the decompressed signal (with possible packet loss), and
4. $e(t) = \hat{f}(t) - \tilde{f}(t)$: the error signal.

II.C.1 Encoded signal at location of loss

First, the attributes of the encoded signal *at the location of the packet loss* are described. For the encoded signal $\hat{f}(t)$, the tendency of human observers to track moving objects with their eyes may enhance visibility of packet loss in smoothly moving regions, yet local signal variance and motion variability may hide the packet loss. Texture masking, luminance masking, and motion masking may each reduce visibility of the packet loss. In a high-quality encoding, these features of the encoded signal are essentially equal to those of the original uncompressed signal. These signal attributes do not depend on the compression standard.

Motion information is considered to be an underlying feature of a video, independent of the compression algorithm. Therefore, the following RR signal descriptors related to motion information directly are measured from the *uncompressed* signal $f(t)$. For each macroblock, its motion vector is measured by forward motion estimation from the previous frame. For each packet, **MOTX** and **MOTY** are the mean motion vectors in the x and y directions over all MBs in the packet. Also, **MotionVarX** and **MotionVarY** are the variances of the the motion vectors in the x and y directions over the macroblocks in the packet. We define a high-motion descriptor **HighMOT** to be true if **MOTM**$= \sqrt{MOTX^2 + MOTY^2} > \sqrt{2}$. **ResidEng** is the average residual energy after motion compensation within a packet. The above motion-related descriptors were also considered in [26]. Finally, **SigMean** and **SigVar** are the mean and variance of the signal $f(t)$ over the MBs in a packet.

II.C.2 Encoded signal surrounding location of loss

The attributes of the encoded signal $\hat{f}(t)$ *surrounding* the location of the packet loss can also affect visibility. For a packet loss *after* a scene cut, the impairments can be masked by the change of the scenes. This is called forward temporal masking and it decreases visibility of packet loss. Backward temporal masking also decreases the visibility of a packet loss *before* a scene cut [29]. In addition, when an entire frame is lost immediately *at* the start of a new scene cut to a still (low motion) scene, even though the still scene will be concealed using a frame from the previous scene, leading to a large MSE, the impairment may be invisible. The low motion in the new scene does not change the displayed images very much, and the new scene may appear to start at the next I-frame [74]. In addition to scene cuts, camera motion is also important to packet loss visibility. Viewers are likely to follow, or track, consistent camera motion. This will enhance the visibility of temporal glitches.

Scene- and reference-related factors were examined in [29] using exploratory data analysis. These factors are extracted from the encoded video signal $\hat{f}(t)$, without losing any accuracy relative to the original uncompressed video. Many techniques exist to detect scene boundaries, including those in [75] and [76]. Each packet loss is labeled by the distance in time between the frame first affected by the packet loss and the nearest scene cut, either before or after. This quantity is **DistFromSceneCut**, and is positive if the packet loss happens after the closest scene cut in display order, and negative otherwise. **DistToRef** per MB describes the distance between the current frame (with the packet loss) and the reference frame used for concealment. This variable is positive if the frame at which the packet loss occurs uses a previous (in display order) frame as reference, and negative otherwise. **FarConceal** is true if **MaxDistToRef** (maximum of |DistToRef| in a slice) ≥ 3. In this inequality, MaxDistToRef has units of frames. A Boolean variable **OtherSceneConceal** is TRUE if $|\text{DistFromSceneCut}| < |\text{MaxDistToRef}|$, where the compared variables must be of the same sign (same direction). In this inequality, the compared variables have units of seconds. If the compared variables have different signs, OtherSceneConceal is FALSE. OtherSceneConceal describes whether the packet loss will be concealed by an out-of-scene reference frame which will increase the visibility of packet loss. To account for the depressed visibility immediately *before* the scene cut, a Boolean variable **BeforeSceneCut** is defined, which is TRUE if $-0.4sec < \text{DistFromSceneCut} < 0sec$ [29]. Depressed visibility *after* a scene cut requires that the packet loss not only appear close to the scene cut, but also *disappear* quickly after the scene cut. Therefore, to account for the depressed visibility immediately after a scene cut, the Boolean variable **AfterSceneCut** is defined, which is TRUE when both OtherSceneConceal is FALSE and $0sec < (\text{DistFromSceneCut} + \text{Duration}) < 0.25sec$.

Camera motion information can also be extracted from the compressed video using a number of techniques, including those in [77]. In this work, scenes are classified based on four camera-motion types: still, panning, zooming, or complex camera motions. Table II.2 indicates the distribution of camera motion both in the complete videos shown to viewers, as well as the fraction of losses which occurred in each type of camera motion. Significantly fewer viewers saw packet loss in still scenes than in panning or zooming scenes. Therefore, **NotStill** is defined to be TRUE if motion type is not still.

II.C.3 Decoded signal

The decoded signal, $\tilde{f}(t)$, at the location of a packet loss has several attributes that affect packet-loss visibility. Due to imperfections in the error concealment of the lost packet, there can be spatial (vertical or horizontal) or temporal discontinuity with the neighboring MBs or frames; these are called *edge artifacts.* A lost frame is likely to introduce temporal edges and a lost slice is likely to introduce both temporal and horizontal edges into the decoded signal. For example, a moving vertical bar that is continuous in the encoded signal may become disjointed in the decoded signal due to the impairment. Vertical edges may also be introduced with FMO, or when the impairment propagates into subsequent frames. All of these edge artifacts are likely to increase the visibility of the impairment. **SBM**, Slice Boundary Mismatch, describes the impact of packet loss on slice boundaries. Methods to measure SBM can be found in [28, 29].

II.C.4 Error signal

The error caused by the impairment, $e(t)$, is completely characterized by its *support* and its *amplitude.* The error support is characterized by spatial support (size, spatial pattern and location) and temporal support (duration). The size is controlled by the packet size as well as the frequency of synchronization codewords like slice start codes. The spatial pattern of the error can be governed by the FMO setting in H.264. The error duration is dominated by the frequency of I-frame or I-block information. The initial amplitude of the error at the time of the loss depends more heavily on the underlying video content and the decoder concealment strategy than on the compression standard itself. The effectiveness of error concealment strategies greatly depends on the content, since some content is more easily concealed than others; however, it can also be improved with a careful selection of encoding parameters. For example, concealment motion vectors in MPEG-2 I-frames are very helpful. The error amplitude may decrease as a function of time even when no I-blocks are present due to the motion-compensation prediction process [78]. In addition, using long-term prediction in H.264 can improve error attenuation [79].

To measure these attributes of the error signal, it is straightforward to extract from a lossy bitstream the exact error size (**SpatialExtent**), spatial pattern, vertical location within the frame (**Height**), and temporal duration (**Duration**). **SXTNT2** is true when two consecutive slices are lost (SpatialExtent=2), **SXTNTFrame** is true when all slices in the frame are lost, and **Error1Frame** is TRUE if the packet loss lasts only one frame (Duration=1).

MSE and SSIM (Structural Similarity Index) are commonly used to characterize the amplitude of the error. For an accurate evaluation of quality degradation due to both compression artifacts *and* packet loss, these must be computed at the encoder, since they depend on $f(t)$. However, we only consider the quality degradation due to packet loss *without* encoding artifacts. Therefore, when calculating MSE and SSIM, $\hat{f}(t)$ is the reference video instead of $f(t)$. As a result, these can be computed at the server.

The MSE directly measures the error due to packet loss, $e(t) = \hat{f}(t) - \tilde{f}(t)$, and is defined for one frame, t, as

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} e_{ij}(t)^2 \tag{II.1}$$

where $M \times N$ is the image video resolution, and i and j are the indexes in the horizontal and vertical directions of the frame. MSE characterizes the error amplitude.

The SSIM for one frame is defined as

$$SSIM(\hat{f},\tilde{f}) = \frac{(2\mu_{\hat{f}}\mu_{\tilde{f}} + C_1)(2\sigma_{\hat{f}\tilde{f}} + C_2)}{(\mu_{\hat{f}}^2 + \mu_{\tilde{f}}^2 + C_1)(\sigma_{\hat{f}}^2 + \sigma_{\tilde{f}}^2 + C_2)} \tag{II.2}$$

where μ and σ are the mean and the standard deviation of the corresponding signal, $\sigma_{\hat{f}\tilde{f}}$ is the cross-correlation coefficient between $\hat{f}$ and $\tilde{f}$, and C_1 and C_2 are constants [1]. SSIM captures the structural statistics of $\tilde{f}(t)$ at the location of the impairment through its mean and variance. However, as with MSE, SSIM characterizes error amplitude but neither error size nor duration. SSIM also does not directly measure the decoded impairment attributes (like horizontal and temporal edges).

Due to the predictive nature of video coding, if a packet is lost, an error may propagate to the predicted frames. To completely describe the error, one must calculate the errors induced on all affected frames. **CumulativeMSE** (**CumulativeSSIM**) is the sum of MSE (SSIM) over all the frames that are affected by a packet loss. To compute these at the encoder, it is necessary to decode once for every single possible packet loss. Thus, accumulating these factors across all affected frames for every possible packet loss dramatically increases the computational complexity. This is prohibitively expensive, and thus neither CumulativeMSE or CumulativeSSIM are considered in the visibility model.

Instead, only the initial error induced by a packet loss within the frame where the packet loss occurs is considered. Two measurements are useful: initial MSE and initial SSIM. These factors can be pooled in two ways. The first is **IMSE** (or **ISSIM**), the MSE (or SSIM) averaged over the entire frame that is initially impacted by the loss. Another pooling strategy for the initial MSE or initial SSIM is to consider extrema over a small spatial window. **MaxIMSE** is defined as the maximum per-MB MSE over all MBs in the initial impairment, and **MinISSIM** is defined as the minimum per-MB SSIM over all MBs in the initial impairment. MaxIMSE was shown to be useful in [27]. An equation to compute a per-MB *initial* SSIM in an RR framework was presented in [28] using the local means and

Table II.2: Distribution of camera motion in videos and in losses

Camera motion type	% Frames	# Losses	% Losses	Mean viewers noticing a packet loss among 12 people
Still	63.7	2380	68.9	1.31
Panning	23.6	814	23.6	3.95
Zooming	6.7	169	4.9	3.99
Complex	1.8	92	2.7	2.62

variances of the encoded and decoded signals, as well as their MSEs. Table II.3 summarizes the factors.

II.D GLM model building approach on multiple data sets

In this section a Generalized Linear Model (GLM) is built on the data sets. The background of GLMs is introduced in Chapter I. For the prediction factors, in order to reduce the dependency on the RR factors sent from the encoder, as discussed in II.B, in this work only the motion and the residual information are used. CumulativeMSE and CumulativeSSIM are considered too computationally intensive, and here the initial MSE and initial SSIM are used.

The subjective datasets available for training the model capture a wide range of possible system configurations: different spatial resolution, compression standards, coding parameters, and error concealment strategies. The RR and NR factors described in Section II.C capture almost all of these variations. For example, the effects of different GOP structures and lengths on packet loss impairment can be partly described by temporal duration of the packet loss, as discussed in Section II.C. The only exception is that neither the encoder nor the quality monitor can know what error concealment strategy will be used by the decoder.

As noted in Table II.1, there is much less subjective data for default concealment than for the other two concealment strategies, and the default concealment produces more visible errors (as indicated in Table II.1 by the mean

Table II.3: Factors for predicting visibility, classified by its attributes, factor types (FR/RR/NR), and whether the factor must be computed at the encoder or can be computed at the server

Factor Attributes	Factor Name	Factor type	Suggested Calculating Point
Signal	SigMean, SigVar, MOTX MOTY, MotionVarX MotionVarY, ResidEng	RR	Encoder
Error	IMSE, ISSIM MaxIMSE, MinISSIM	RR	Server
	SpatialExtent, Duration, Height	NR	Server
Scene	DistFromSceneCut BeforeSceneCut, AfterSceneCut	RR	Server
Concealment reference	DistToRef, OtherSceneConceal FarConceal	RR	Server
Camera motion	NotStill	RR	Server

number of viewers who saw each loss). There is most data for the MCEC, which produces fewer noticeable errors. If the model were trained using samples chosen randomly from the combined dataset, the resulting fit will be dominated by the MCEC strategy. Therefore, models are trained using an equal number of samples from each of the datasets, and then use cross-validation to evaluate the goodness of fit and select the best model. Cross-validation [80] is commonly used for model evaluation and to prevent over-fitting when data is sparse. A model is trained on a fraction of the data (*training set*) and then tested using the remaining data points (*testing set*). A partition like this is known as a *fold*, and it is repeated for different folds with different training and testing partitions of the data. The training and testing sets are to achieve equal representation from all datasets including Dataset 1, which has the fewest samples (215). Specifically for each fold, 159 samples from each dataset are randomly chosen to fit a model using 159×3 training data. Also, a testing set contains the remaining 56 samples from Dataset 1, the remaining 921 samples from Dataset 2, and the remaining 2001 samples from Dataset 3. The model coefficients are estimated from the *training set* for given factors, and then we evaluate the performance error of the fitted model in the jth fold using the *testing set* as follows:

$$q_j = \frac{1}{3}\sum_{k=1}^{3}\left[\frac{1}{N_k}\sum_{\substack{i\text{th packet loss}\\ \text{in testing set } k}}(p_i - \tilde{p}_i)^2)\right], \tag{II.3}$$

where $\tilde{p}_i$ is the predicted fraction of viewers who saw the i^{th} packet loss, and N_k is the number of samples in the testing set of Dataset k. Four-fold cross-validation is performed: the fitting process is run for a total of four times with four different folds, therefore producing 4 fitted models and $q_j, j = 1, 2, 3, 4$. This four-fold procedure is repeated four times with four different random seeds. The average performance error of these sixteen models is Q.

$$Q = \frac{1}{16}\sum_{r=1}^{4}\sum_{j=1}^{4}q_j^r \tag{II.4}$$

Table II.4: Factors in the final model

Factors	Coeff. for Final Model
Intercept	4.18061
$\log(1 - \text{ISSIM} + 10^{-7})$	0.22871
SXTNT2	-0.41208
SXTNTFrame	-1.47672
Error1Frame	-0.33009
$\log(\text{MaxIMSE} + 10^{-7})$	0.27578
$\log(\text{ResidEng} + 10^{-7})$	-0.61219
HighMOT	0.18290
NotStill	0.73364
BeforeSceneCut	-1.14434
OtherSceneConceal	2.08966
$\log(\text{IMSE} + 10^{-7})$	0.30492
$\log(\text{IMSE} + 10^{-7})\times$ FarConceal	0.25720

where the superscript r stands for the r^{th} random seed.

For factor selection, Q is used to decide if a specific factor is significant and should be included in the model: for each considered factor added to the model, Q is calculated by the 4-seeds-4-folds GLM modeling process. A factor is included only if the model with that factor included has smaller Q than the model without that factor. By the same idea, factors are excluded from the model if it has lower Q without them. To obtain the factor coefficients, fitting is used from the seed that achieved the lowest performance error. The factors and coefficients of the final model are summarized in Table II.4. Since the model is developed based on data from different GOP types, and the factors are not GOP-type-specific, this packet loss visibility model is versatile enough to be applied to video compressed with various GOP types.

II.E Conclusion

In this chapter, we discuss a generalized linear model for packet loss visibility applicable to different GOP structures. The contributions of this chapter are that, unlike earlier models, this visibility model is developed on datasets from multiple subjective experiments using different codecs, different encoder settings, and different decoder error concealment strategies. So the model has broad applicability.

II.F Acknowledgements

This work was supported in part by the National Science Foundation, the Center for Wireless Communications at UCSD and the UC Discovery Grant Program of the State of California.

Chapter II of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman and A. 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, March 2010. I was the primary author and the co-authors Prof. Cosman and Dr. Reibman directed and supervised the research which forms the basis for Chapter II. The co-authors Dr. Kanumuri and Dr. Poole also contributed to the ideas in this work. The co-author Y. Zhi helped the simulation process.

III

Packet Prioritization using the encoder-based model

The visibility model described in Chapter II allows an encoder to estimate the importance of each individual compressed video packet. How can we exploit this estimate? In Chapter IV, we will use the estimate for unequal error protection. In the current chapter, we apply the visibility model from Chapter II to packet prioritization for a video stream. Before we send video packets into a lossy network, we estimate the packet loss visibility using the encoder-based model. Based on the values, we insert high or low prioritization bits into packets. Using the packet priorities, during network congestion, an intermediate router can intelligently drop low-priority packets.

Several previous works store cumulative MSE information as a "hint track" for the packet, and the router, by inspecting the information, can choose to drop packets with lower distortion [30, 31, 32, 35, 36]. We compare our visibility-based packet prioritization strategy with the Cumulative-MSE-based prioritization method and the widely-used Drop-Tail policy using NS-2 (Network Simulator) [39]. This chapter demonstrates the utility of the model in a general packet prioritization application; the application uses various GOP structures, including ones which were not used in the subjective experiments used for building the model. The comparisons are made using the well-known perceptual quality metric VQM [4].

This chapter is organized as follows: Section III.A presents the design for the packet dropping experiment. Section III.B presents the experiment results comparing our visibility-based prioritization method with others. Section III.C concludes the chapter.

III.A Experimental design

The generalized-GOP visibility model can be used in different applications, such as packet prioritization, unequal error protection and network quality monitoring. In this section we present how the visibility model can be used to prioritize packets, and how this prioritization scheme helps an intermediate router in a congested network decide which packets should be dropped to minimize degradation in the quality of the transmitted video stream. In particular, we demonstrate that while the visibility model was designed for high-quality video over a mostly reliable network, it is still applicable when the video is more heavily quantized and there are more packet losses.

Major applications of packet classification are *packet prioritization for a differentiated-services network* [30, 31, 32], *packet scheduling in the transmitter* [33, 34] in which video packets are sent/resent by an optimal schedule based on the packet classification, transmission delay and network status etc., and *packet discarding at an intermediate router* [35, 36], in which packets are discarded, in the event of network congestion, based on packet classification and bandwidth of the outgoing link from the router. In particular, an optimization algorithm is developed in [36] to be run in the router to optimally discard less important packets. However, it is technically difficult to implement complex algorithms (such as rate-distortion optimization) in current intermediate routers. Also for all the methods mentioned above, the algorithms utilize the cumulative MSE, which is computationally expensive to measure since it includes the MSE due to error propagation.

Therefore, our aim is to develop an efficient packet dropping policy for the router. We propose the *perceptual-quality based packet prioritization* policy, denoted PQ, designed by the visibility model that prioritizes packets. At the server, we set a packet to be low priority when its visibility is less than 0.25 as predicted by the model, and high priority otherwise. The 1-bit high/low priorities can be signaled in the packet itself. The router can be therefore designed to drop packets of low priority to reduce traffic during network congestion. The intermediate router with this capability is realizable in a DiffServ (Differentiated Services) network [81]. Furthermore, instead of the cumulative MSE, the *initial* MSE, which only considers errors in a frame in which the packet is lost, is used for the factor consideration.

The Hint Track method in [35, 36] can not directly be used as a basis of comparison for our method. On the one hand, we consider their optimization algorithm too complicated to run in today's routers. On the other hand, the packet dropping policy in [35, 36] can not be entirely implemented at the server which has a much better computational ability, since it uses knowledge of the router's instantaneous outgoing bandwidth, which is not accessible to the server. However, we can compare our algorithm with their notion of using cumulative MSE. A one-bit prioritization scheme, called cMSE (Cumulative MSE prioritization method), is designed. The cumulative MSE for a particular packet is measured by summing the MSE in all frames in a video affected by the packet drop, and a packet is assigned high priority if the cumulative MSE due to its loss is larger than a threshold, and low priority otherwise. The threshold is derived such that we have approximately the same number of high-priority packets for both cMSE and PQ prioritization. We also compare to the Drop-Tail (DT) policy, a widely-implemented packet dropping approach, which drops packets at the end of the buffer queue in the router when the network is congested. The different policies are evaluated based on the received video quality, measured by VQM.

We simulate the experiment using NS-2 [39] for a network topology shown

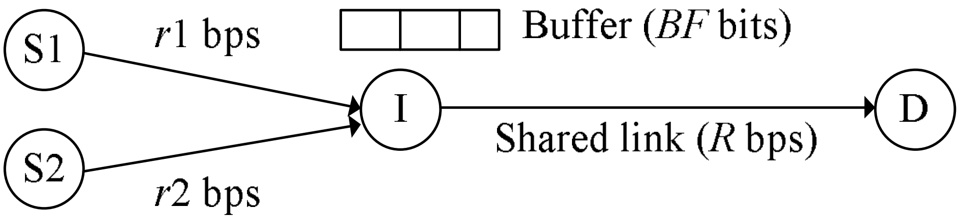


Figure III.1: Topology of experimental network

in Fig. III.1. Two videos (variable-bit-rate encoded at $r1$ and $r2$ bps on the average) are transmitted simultaneously from sources S1 and S2 to destination D. Packets belonging to both videos compete for space in the queuing buffer (of size BF bits) at intermediate node I. The bottleneck link's bit-rate is constant at R bps. When instantaneous rates of S1 and S2 sum to more than R, packets accumulate in the buffer. If this condition persists, the buffer will eventually overflow and packets are dropped in accordance with a policy. At destination D, the quality of received videos is evaluated using VQM, which ranges from 0 (excellent quality) to 1 (poorest possible quality).

Six videos (two videos for each motion type - still, low and high motion) of 10s duration are coded at $R/2$ bps using the H.264/AVC JM codec, with MCEC implemented in the decoder of the server and the receiver. Each simulation with a pair of source videos produces a pair of corresponding received videos for each policy. We form 9 pairs from the six videos such that a balanced representation (each type of video competes twice with all the three types) is obtained. For each of the received videos (18 from 9 pairs), a policy wins if its VQM score is lower than the other policy used for comparison, and a tie occurs when the policies have identical VQM scores. This procedure is repeated for each (R,BF) setting of interest. To show the effectiveness of our policy across different GOP structures, we conducted experiments with *IPPP*, *IBBP* and *Pyramid* (Fig. III.2) encoding structures, and the numbers of reference frames are 1, 2 and 4 respectively. I-frames are repeated every 24 frames for all three of these different GOP structures.

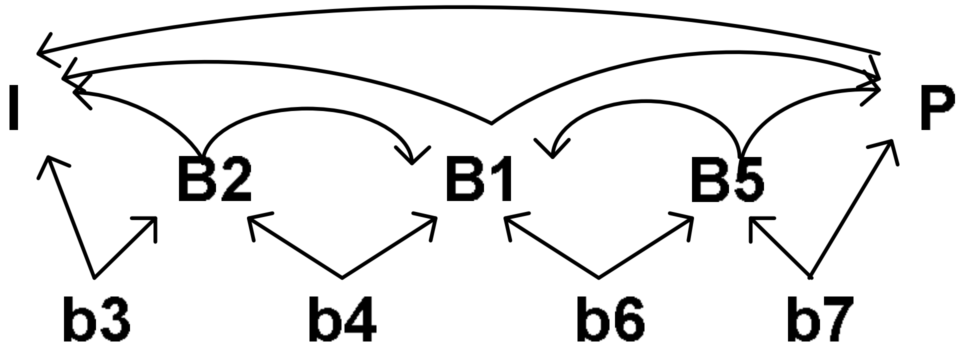


Figure III.2: Pyramid GOP structure; A B-frame in upper case can be used for reference while the ones in lower case can not. The I frame is coded first, the P frame is coded second, and then the numbers indicate the coding order within the group of B/b frames.

III.B Experimental results

III.B.1 PQ comparison with DT and cMSE

Table III.1 shows the comparison results for different buffer sizes when the bottleneck rate is fixed at R=1200 kbps. A larger buffer size is used for Pyramid because the effect of out-of-display-order coding is more prevalent than the other two GOP structures and hence its bitstream is burstier. To quantify the performance comparison, we define $comparison\ ratio = \sum \#wins / \sum \#losses$ for each GOP-Competitor comparison. The proposed PQ prioritization significantly outperforms DT with comparison ratios of 5, 2 and 1.27 for IPPP, IBBP and Pyramid, respectively. We can observe from the table that this trend occurs across all settings of buffer size. When compared with cMSE, the proposed method has a large advantage for Pyramid and IBBP with comparison ratios of 6.14 and 5 respectively. However, in the case of IPPP, the proposed method has a slight disadvantage (comparison ratio of 0.687) when compared with cMSE. The average of the comparison ratios from the six GOP-Competitor comparisons is as high as 3.34, which means on average we perform considerably better than the other policies. We also did similar experiments at a lower fixed bottleneck rate (R=800 kbps) and the results can be seen in Table III.2. We observe a similar trend as in Table III.1 (we win for 5 GOP-Competitor comparisons and lose for 1). We continue to have a good average of comparison ratios (2.01) although it is lower than that in Table III.1 (3.34). The reason we have a lower average of comparison ratios for a lower fixed bottleneck rate could be the fact that the data used for building the model were collected from videos with no obvious coding artifacts, but the model is being applied to videos at lower rates which do have obvious coding artifacts. However, the model is still capable of prioritizing the video well and outperforms other policies.

Table III.3 compares the performance of the different policies for a variety of bottleneck rates while the buffer size is fixed. An important observation, again, is that we perform relatively better at a higher encoding rate (R=1200 kbps) than at lower rates. Nevertheless, the performance of the model is quite robust at lower encoding rates. For IBBP, the proposed PQ prioritization performs very well at all encoding rates, and the comparison ratios are 1.57 over DT, and 3.9 over cMSE. For Pyramid, we have a good comparison ratio over cMSE (2.17), while the comparison ratio is smaller (1.07) when compared with DT. For IPPP, we outperform DT with a comparison ratio of 4.40, but we lose slightly against cMSE (0.741). Table III.4 shows very similar comparison results for a higher fixed buffer size. The average of comparison ratios for these two tables remains almost the same (2.31 for Table III.3 and 2.39 for Table III.4). This shows that we consistently perform better than other policies across different fixed buffer sizes.

From Tables III.1, III.2, III.3 and III.4, an interesting observation is found: for videos of Pyramid and IBBP, PQ outperforms cMSE even more than it outperforms DT. This is interesting because DT is a simplistic policy with no consideration of video content, and one would expect a policy that takes video content into account to do better. To understand why DT does as well as it does, we compared DT, which drops tail packets during congestion, against DropRandom (DR), which randomly drops any buffered packet during congestion. For all network conditions, DT outperformed DR for those GOP structures which have B frames (comparison ratio $= \sum \#wins / \sum \#losses = 12.81$ over all cases in Pyramid, and 5.42 in IBBP), and it did worse (comparison ratio = 0.53 over all cases) for the GOP structure (IPPP) which has no B frames. To explain the better performance of DT than DR in GOP structures with B frames, let us consider the IBBP structure. The encoder must set aside the two B frames in order to encode the P frame, and then it can encode the two B frames. Assuming frame encoding time is much smaller than frame display time, we can consider that the encoder releases all the bits at once, corresponding to the P frame followed by the two B frames. The router queueing buffer is therefore sitting with B frame bits at the tail. After the encoder waits for the next three frames, it processes and releases their bits all at once, again the router queueing buffer will be sitting with B frame bits at the tail. The DT policy almost always finds B-frame packets at the tail. Dropping B frame packets is of course desirable because there is no error propagation. This is the reason why DT can perform well and does better than DR for the GOP structures with B frames (IBBP or Pyramid). The advantages of DT in IBBP and Pyramid can overtake cMSE even though cMSE is much better than DR in every case of our network scenarios and GOP structures (the comparison ratio=5.60 in Pyramid, 3 in IBBP and 3.34 in IPPP). However, the advantages of DT in IBBP and Pyramid are not enough to overtake the visibility-based prioritization.

The PQ prioritization works well in most of the cases (five out of six GOP-Competitor comparisons) in each of the four tables. In particular, the proposed policy is always better than DT, a widely implemented dropping method in existing intermediate routers, and is better than cMSE for two out of three cases. The reason that we are not better than cMSE in IPPP is that all frames in this GOP structure are reference frames. Hence an important factor, Error1Frame (indicating whether the loss last only for one frame), in the model is the same for all frames and can not be used to distinguish the importance of a packet. Therefore, in IPPP, we perform slightly worse than cMSE. However, cMSE is a very computationally expensive approach, since it is based on cumulative MSE which has to account for the MSE due to error propagation. Instead of cumulative MSE, the visibility model just uses initial MSE (MSE in the frame where the packet loss occurred) which is computationally trivial.

Another performance comparison is illustrated in Figure III.3, where average VQM scores among PQ, DT and cMSE over the competitions in Tables III.1, III.2, III.3 and III.4 are shown. We can see that on average, the VQM scores obtained by PQ are lower (better) than those by DT and cMSE in different comparison scenarios. We conclude that our PQ prioritization not only improves more cases on video quality most of the time, as shown in Tables III.1 through III.4, but also on average has lower (better) VQM scores over different comparisons.

Although the visibility model is built using data of isolated losses (one packet loss for every four seconds, as discussed in chapter II), the model is quite robust to different packet loss rates in the simulations for real networks. In these experiments, depending on the buffer size and the transmission rate and the variability of the video content, packet losses occur with different degrees of bursty behavior. The model does well consistently across different buffer sizes and transmission rates. Note that in our experiments, the buffer sizes are chosen such that the packet loss rate has a reasonable range for video quality; our packet loss rates (0.7%-20%) are similar to those investigated in the literature (e.g., [18, 20]).

III.B.2 Packet loss rate for PQ, cMSE and DT

In addition to VQM score comparisons for various network conditions detailed in subsection III.B.1, we also analyze the packet loss rate (PLR) induced by each of the three dropping policies (PQ, DT and cMSE) for Pyramid, IBBP, and IPPP in Figure III.4. A PLR value corresponding to a dropping method in a

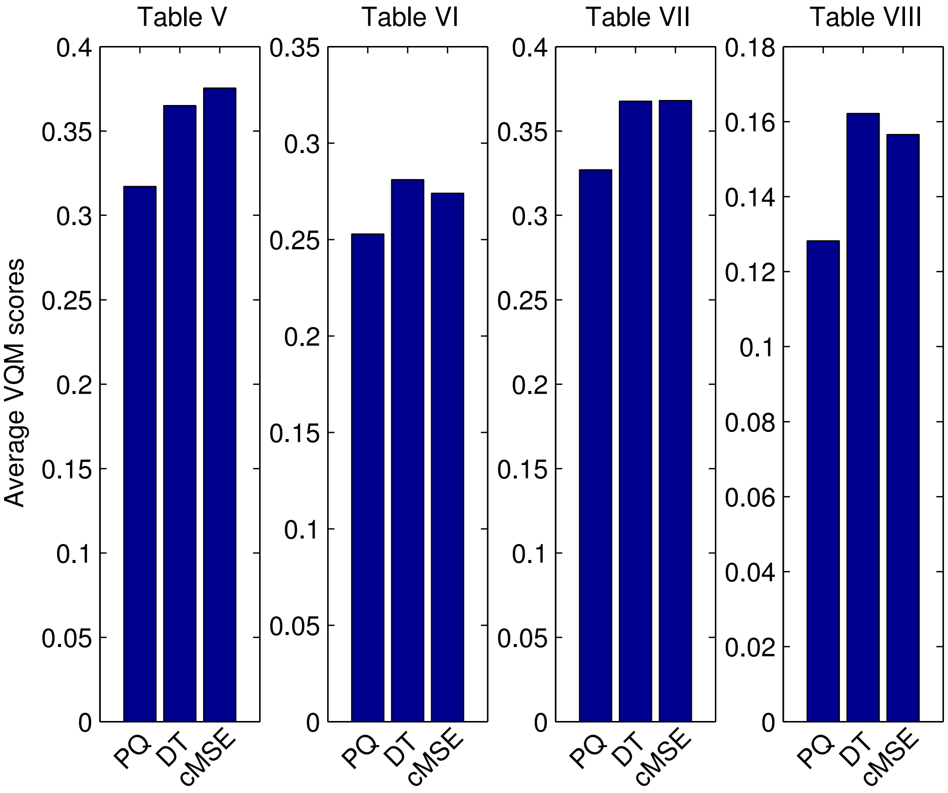


Figure III.3: Comparisons of average VQM scores among PQ, DT and cMSE over the competitions in Tables III.1, III.2, III.3 and III.4

Table III.1: Proposed PQ compared to DT and cMSE: Higher Fixed bottleneck rate (R kbps) and varied buffer size (BF kbits). Average of comparison ratios=3.34. W stands for Wins, L for Losses and T for Ties.

Pyramid (R=1200)				**IBBP** (R=1200)				**IPPP** (R=1200)			
vs. **DT** (comp. ratio=1.27)	W	L	T	vs. **DT** (comp. ratio=2)	W	L	T	vs. **DT** (comp. ratio=5)	W	L	T
BF=200	10	8	0	BF=80	12	6	0	BF=80	18	0	0
BF=400	9	9	0	BF=100	12	6	0	BF=100	13	5	0
BF=600	9	5	4	BF=120	12	6	0	BF=120	14	4	0
vs. **cMSE** (comp. ratio=6.14)	W	L	T	vs. **cMSE** (comp. ratio=5)	W	L	T	vs. **cMSE** (comp. ratio=0.68)	W	L	T
BF=200	15	3	0	BF=80	16	2	0	BF=80	9	9	0
BF=400	16	2	0	BF=100	14	4	0	BF=100	7	11	0
BF=600	12	2	4	BF=120	15	3	0	BF=120	6	12	0

GOP is obtained by averaging the PLRs from corresponding (R, BF) pairs listed in the tables from subsection III.B.1. Figure III.4 shows that the proposed PQ prioritization drops slightly more packets than DT or cMSE on average. In spite of the higher PLR values, our PQ performs well as shown in subsection III.B.1. Also in each comparison, the bit-rate for the bottleneck link is the same for the compared policies. Therefore, with higher PLR by PQ, we infer that the average size of dropped packets with PQ is smaller than that of other policies. Our PQ drops slightly more packets, but they are smaller-size visually unimportant packets, and therefore PQ achieves a better perceptual video quality. This result also indicates that traditional video quality assessments based on the PLR may not relate well to perceptual video quality.

III.C Conclusion

We use the visibility model from Chapter II to prioritize video packets and discard packets based on perceptual quality. Experiments done under diverse

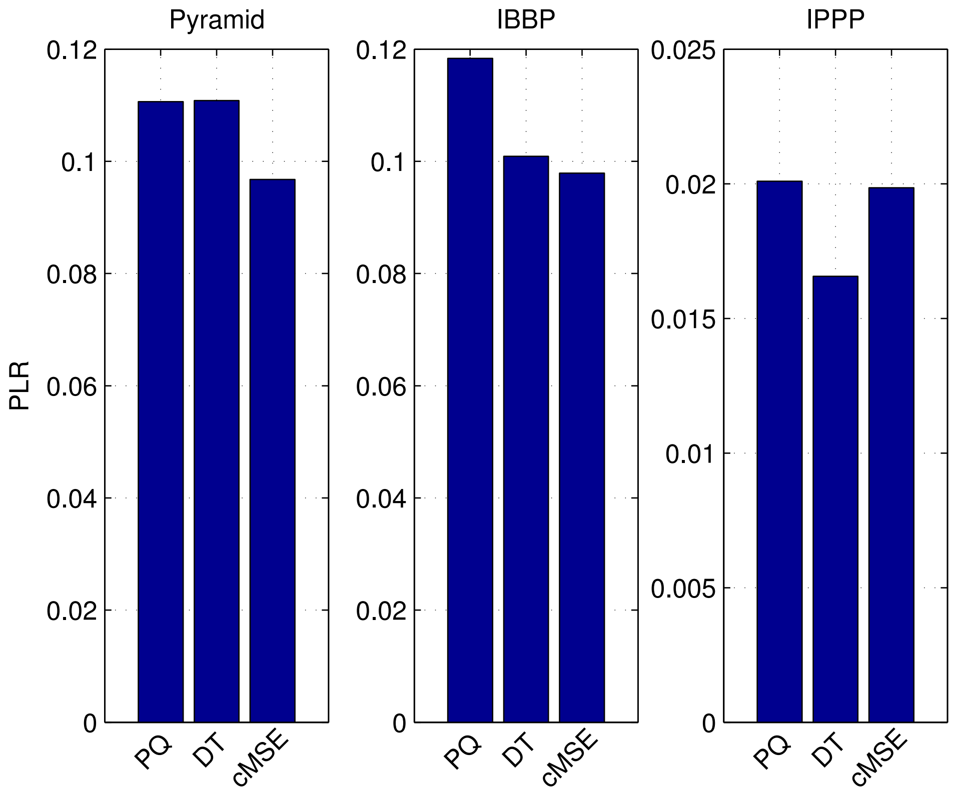


Figure III.4: Average PLR (Packet Loss Rate) comparisons among PQ, DT and cMSE in different GOP structures.

Table III.2: Proposed PQ compared to DT and cMSE: Lower Fixed bottleneck rate (R kbps) and varied buffer size (BF kbits). Average of comparison ratios=2.01. W stands for Wins, L for Losses and T for Ties.

Pyramid (R=800)				**IBBP** (R=800)				**IPPP** (R=800)			
vs. **DT** (comp. ratio=1.21)	W	L	T	vs. **DT** (comp. ratio=1.57)	W	L	T	vs. **DT** (comp. ratio=3.5)	W	L	T
BF=200	10	8	0	BF=80	11	7	0	BF=80	13	5	0
BF=400	7	7	4	BF=100	11	7	0	BF=100	15	3	0
BF=600	6	4	8	BF=120	11	7	0	BF=120	14	4	0
vs. **cMSE** (comp. ratio=2.5)	W	L	T	vs. **cMSE** (comp. ratio=2.85)	W	L	T	vs. **cMSE** (comp. ratio=0.459)	W	L	T
BF=200	13	5	0	BF=80	14	4	0	BF=80	7	11	0
BF=400	10	4	4	BF=100	14	4	0	BF=100	5	13	0
BF=600	7	3	8	BF=120	12	6	0	BF=120	5	13	0

network conditions and GOP structures show that the PQ dropping performs better than the policy using cumulative MSE as used in the Hint-Track method in most cases, and outperforms the widely-implemented Drop-Tail in all cases. Although the model is designed for high-quality video transported over a mostly reliable network, the experiments show that the model performs well for videos with various encoding rates. The analysis on packet loss rate across three different dropping policies shows that our policy achieves a better visual quality by dropping more, but perceptually unimportant, packets with smaller sizes. This emphasizes that evaluating video quality based solely on packet loss rate is inaccurate.

III.D Acknowledgements

This work was supported in part by the National Science Foundation, the Center for Wireless Communications at UCSD and the UC Discovery Grant Program of the State of California.

Chapter III of this dissertation, in part, is a partial reprint of the ma-

Table III.3: Proposed PQ compared to DT and cMSE: Lower Fixed buffer size (BF kbits) and varied bottleneck rate(R kbps). Average of comparison ratios=2.31. W stands for Wins, L for Losses and T for Ties.

Pyramid (BF=300)				**IBBP** (BF=80)				**IPPP** (BF=80)			
vs. **DT** (comp. ratio=1.07)	W	L	T	vs. **DT** (comp. ratio=1.57)	W	L	T	vs. **DT** (comp. ratio=4.4)	W	L	T
R=800	9	9	0	R=800	11	7	0	R=800	13	5	0
R=1000	9	9	0	R=1000	10	8	0	R=1000	13	5	0
R=1200	10	8	0	R=1200	12	6	0	R=1200	18	0	0
vs. **cMSE** (comp. ratio=2.17)	W	L	T	vs. **cMSE** (comp. ratio=3.9)	W	L	T	vs. **cMSE** (comp. ratio=0.74)	W	L	T
R=800	13	5	0	R=800	14	4	0	R=800	7	11	0
R=1000	13	5	0	R=1000	13	5	0	R=1000	7	11	0
R=1200	11	7	0	R=1200	16	2	0	R=1200	9	9	0

terial as it appears in T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman and A. 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, March 2010. I was the primary author and the co-authors Prof. Cosman and Dr. Reibman directed and supervised the research which forms the basis for Chapter III. The co-authors Dr. Kanumuri and Dr. Poole also contributed to the ideas in this work. The co-author Y. Zhi helped with the simulation process.

Table III.4: Proposed PQ compared to DT and cMSE: Higher Fixed buffer size (BF kbits) and varied bottleneck rate(R kbps). Average of comparison ratios=2.39. W stands for Wins, L for Losses and T for Ties.

Pyramid (BF=600)				**IBBP** (BF=140)				**IPPP** (BF=140)			
vs. **DT** (comp. ratio=1.5)	W	L	T	vs. **DT** (comp. ratio=2.6)	W	L	T	vs. **DT** (comp. ratio=3)	W	L	T
R=800	6	4	8	R=800	13	5	0	R=800	14	2	2
R=1000	3	3	12	R=1000	11	7	0	R=1000	13	5	0
R=1200	9	5	4	R=1200	15	3	0	R=1200	12	6	0
vs. **cMSE** (comp. ratio=3.28)	W	L	T	vs. **cMSE** (comp. ratio=3.5)	W	L	T	vs. **cMSE** (comp. ratio=0.5)	W	L	T
R=800	7	3	8	R=800	13	5	0	R=800	7	8	3
R=1000	4	2	12	R=1000	14	4	0	R=1000	5	13	0
R=1200	12	2	4	R=1200	15	3	0	R=1200	5	13	0

IV

RCPC rate allocation for perceptual video quality

For videos transmitted in an error-prone network, it is necessary to protect the source bitstream. The sender can perform unequal error protection against the bit errors from a wireless channel [46, 47, 48]. The research area in joint source and channel coding has been growing rapidly [40, 41, 42, 43, 44, 45]. Joint source and channel coding for progressively coded images/videos are studied in [52, 53]. To stop error propagation due to packet losses, combined consideration of channel coding, source coding and intra updating intervals is necessary [49, 50, 51].

Most of the work concentrates on the situation where the video encoder is scalable, and the measurement for packet importance is the MSE induced by the packet loss. Here we consider non-scalable video streams pre-encoded and stored. Also in this chapter, to correlate better with human perception, we use the encoder-based packet loss visibility model from Chapter II to measure the perceptual importance of each packet. We aim to optimally allocate rate-compatible punctured convolutional (RCPC) codes to minimize the visual quality degradation when transmitting video over an AWGN channel, given a total rate budget. With various packet sizes and distortions for each packet, we transform the original problem into a binary-decision problem. This integer programming problem is first solved by the Branch and Bound method (BnB) [54]. The complexity of the algorithm can be reduced by k-means clustering before using the BnB [56]. The lower complexity algorithm can still provide comparable results. We next use the subgradient method to search in the dual domain for the optimal RCPC channel code rate allocation for each packet. The complexity is further lowered and we can consider the full 13 RCPC codes and do not need to perform packet grouping to obtain approximate solutions as we did. We also exploit the advantage of not sending or not coding packets of lower importance. This boosts the performance especially in worse channel conditions.

The idea of not channel coding and not sending the packets was also used in [57], however, they consider a packet erasure channel, and optimal RS erasure coding. Our work treats the same problem of optimal code rate allocation for already encoded packets, but we consider a bit error channel and RCPC codes. The problem is therefore to unequally protect the packets for a given outgoing channel bit rate and channel SNR. The coding decision is based on both the distortion induced by the packet, as well as the size of the packet. We will compare our algorithms with one from [57].

The organization of this chapter is as follows. Section IV.A formulates the RCPC rate allocation problem as an integer programming problem. In Section IV.B, we discuss several different algorithms to find the solution. Section IV.C discusses the simulation results. Section IV.D compares our algorithm with the method from [57]. Section IV.E concludes the chapter.

IV.A RCPC rate allocation for expected packet loss visibility

Using the model described in Chapter II, we can compute for each packet the loss visibility. The visibility scores can be regarded as the visual importance of each packet. If we assign a lower channel code rate to the packet, the probability that the packet will be lost is lower. However, the lower code rate requires more FEC (Forward Error Correction) bits. Thus it is important to find out the best code rates to be allocated to each packet given a total bit budget and other characteristics of the packets.

Assume we have N packets, where the ith packet has size S_i and packet loss visibility V_i, $i = 1, 2, ..., N$. We seek the optimal RCPC rate r_i for the ith packet from the RCPC candidate set $\{R_1, R_2, ..., R_K\}$, so as to minimize the end-to-end expected packet loss visibility, while the outgoing total rate budget is constrained to be B. The packet error probability P_e depends on channel SNR, packet size, and RCPC code rate selected for the packet. Each packet will be appended with a 16-bit CRC for error detection. We include the CRC bits in the packet size and assume perfect error detection. Whenever there is at least one bit error in the packet after channel decoding, we discard the packet. Therefore, $P_e = 1 - (1 - P_b(SNR, r_i))^{S_i}$ where $P_b(SNR, r_i)$ (hereafter denoted P_b) is the bit error probability after channel decoding for code rate r_i. The optimization problem of minimizing the expected packet loss distortion subject to the total bit constraint can be formulated as:

$$\min_{\mathbf{r}} \sum_{i=1}^{N} V_i \{1 - (1 - P_b)^{S_i}\} \text{ subject to } \sum_{i=1}^{N} \frac{S_i}{r_i} \leq B$$
$$r_i \in \{R_1, R_2, ..., R_K\}, \quad i = 1, 2, ..., N \tag{IV.1}$$

where $\mathbf{r} = [r_1, r_2, ..., r_N]$.

This problem can be recognized as a nonlinear integer programming problem, and we will discuss several different algorithms to solve this problem.

IV.B Solutions to the integer programming problem

IV.B.1 Branch and bound method

The problem can be solved by a well-developed method called branch-and-bound (BnB). To use BnB to solve an integer optimization problem, it is preferable to transform the original discrete optimization variables into binary boolean variables [82]. For our problem, the optimization variables and the set $r_i \in \{R_1, R_2, ..., R_K\}$ are transformed into $x_{ij} \in \{0, 1\}$ defined as

$$x_{ij} = \begin{cases} 1 & \text{if packet } i \text{ uses rate } j \\ 0 & \text{otherwise} \end{cases}$$

Since one packet can only use one rate, we have the following linear equality constraint:

$$\sum_{j=1}^{K} x_{ij} = 1, \forall i = 1, 2, ..., N$$

Therefore, our problem in (IV.1) can be written as:

$$\begin{aligned} \min_{x_{ij}} \quad & \tfrac{1}{N} \textstyle\sum_{i=1}^{N} \sum_{j=1}^{K} x_{ij} \times V_i \times \{1 - (1 - P_b)^{S_i}\} \\ s.t. \quad & \textstyle\sum_{i=1}^{N} \sum_{j=1}^{K} x_{ij} \times S_i \times \frac{1}{R_j} \leq B \\ & \textstyle\sum_{j=1}^{K} x_{ij} = 1, \forall i = 1, 2, ..., N \\ & x_{ij} \in \{0, 1\} \\ & i = 1, 2, ..., N, j = 1, 2, ..., K \end{aligned} \tag{IV.2}$$

BnB partitions the original problem into smaller subsets by tree-growing, and eliminates further consideration of the feasible solutions that can not be better than the current one [82]. The algorithm solves the binary-variable problem as follows :

1. The original problem is solved with the integer constraint relaxed to allow $0 \leq x_{ij} \leq 1$. A lower bound (LOWER) of the optimized value to the original problem is the optimized value of the relaxed problem since it has a larger feasible set. An upper bound (UPPER) of the optimized value to the original problem can be obtained by substituting the rounded solutions (to zero or one so that they are feasible) into the problem. This is an upper bound of the optimal value to the original problem since it is the best feasible value we can find at this stage. The optimal value to the original problem must be less than or equal to UPPER, and greater than or equal to LOWER.

2. For non-zero-or-one entries (which are infeasible) in the solution from the previous stage, the algorithm grows binary subtrees that fix one entry to zero or one, then solves the problem while relaxing other entries. The optimized value is a lower bound to this subproblem. If this lower bound is greater than the UPPER, we prune this branch, since the solutions to any feasible combination of the trees growing from this point are not going to be better (less) than the UPPER we currently have. Otherwise, we keep the node for further growing. The upper bound of this subproblem can again be yielded by substituting the rounded solution to the problem. If the upper bound to this subproblem is less than UPPER, we update UPPER, and mark this rounded solution (feasible) as the best solution so far.

3. Step 2 will be repeated until there is no node to be grown.

Details of the BnB algorithm can be found in [82].

In our simulation, we have N=450= 15 packets/frame × 30 frames/GOP for our experiment. If the optimization is done over one frame, the algorithm is not able to exploit the relative differences in visibility that occur in different frames and redistribute FEC bits efficiently. For example, if we optimize the channel rate allocation over one I frame where the loss visibility of most packets would be high, we are not making use of the less necessary FEC bits used in B frames where there are more packets with lower visibility. So ideally, we should perform BnB on all packets in a GOP using all ($K = 13$) RCPC codes for the algorithm to select from. However, if the optimization is to be done over a GOP (30 frames), there will be $N = 450$ variables, which is too large for the BnB method; the complexity of BnB has worst-case exponential time [55].

Here we consider all packets in a GOP, but we partition the packets into groups. We first sort the packets by their packet loss visibility in ascending order. Each group of packets ($N = 15$) to be optimized by BnB includes 13 packets of low visibility from the head of the sorted packets, and 2 packets of high visibility from the tail of the sorted packets. For example, after sorting the packets by the estimated loss visibility in ascending order, the first group of packets includes packets 1-13, 449, 450 from the sorted packets, and the second 14-26, 447, 448, etc. This way, we attempt to distribute FEC bits from more packets of low visibility, to few packets of high visibility. Also due to complexity, we only use 4 channel code rates to select from ($\{\frac{8}{12}, \frac{8}{14}, \frac{8}{16}, \frac{8}{18}\}$). Therefore for each GOP, we need to perform a (N,K)=(15,4) BnB 30 times. We denote this the **SORTED-4** algorithm.

IV.B.2 K-means clustering

An algorithm involving (N,K)=(15,4) BnB 30 times for a GOP may still be complicated for real-time processing, therefore, we aim to reduce the complexity of the algorithm. From Equation IV.1 we see that the optimization problem depends on not only the visibility of the packet (V_i), but also the size of the packet (S_i). The idea is to group packets of similar visibility and packet size together and consider this group as one packet. Thus we propose to use k-means clustering to group packets before using BnB. We consider N packets in a GOP and each has a 2-dimensional vector (V_i,S_i). The k-means clustering algorithm partitions the N vectors into P clusters with the goal of minimizing the sum, over all clusters, of the within-cluster sums of point-to-cluster-centroid distances. Each of the P clusters has a quantized vector ($\bar{V}_z$,$\bar{S}_z$) for the cluster, $z = 1, 2, ..., P$. We then use these P vectors to perform the BnB. Equation IV.1 can now be rewritten as:

$$\begin{aligned}\min_{\bar{\mathbf{r}}}\ & \tfrac{1}{N}\textstyle\sum_{z=1}^{P} \bar{V}_z \times Num_z \times \{1-(1-P_b)^{\bar{S}_z}\}\\ s.t.\ & \textstyle\sum_{z=1}^{P} \bar{S}_z \times Num_z \times \tfrac{1}{\bar{r}_z} \leq B\\ & \bar{r}_z \in \{R_1, R_2, ..., R_K\}\\ & z = 1, 2, ..., P\end{aligned} \tag{IV.3}$$

where $\bar{\mathbf{r}} = [\bar{r}_1, \bar{r}_2, ..., \bar{r}_P]$, and Num_z is the number of vectors in cluster z. After solving this problem, the optimal rate allocation for each packet $i, i = 1, 2, ..., N$, is the optimal $\bar{r}_z$ where packet i is in cluster z. In our experiment, we use $P =$ 15. Therefore, for each GOP, we only need to do one $(N = P, K)$=(15,4) BnB, which is 30 times less complicated than SORTED-4 is. The proposed algorithm is denoted **KMEANS-4**. We also demonstrate the performance of this algorithm for $(N = P, K)$=(15,6) using 6 channel code rates $\{\frac{8}{10}, \frac{8}{12}, \frac{8}{14}, \frac{8}{16}, \frac{8}{18}, \frac{8}{20}\}$, denoted **KMEANS-6**.

IV.B.3 Subgradient method

For the above two methods, heuristic packet groupings were used to reduce complexity, and only 4 or 6 RCPC codes were used [54, 56]. In this subsection, we use a low-complexity subgradient search in the dual domain to efficiently find the best code for each individual packet from the full set of 13 RCPC rates. We also discuss and utilize the advantage of discarding packets.

We first relax our constrained optimization problem in (IV.1) to an unconstrained problem [83, 84]. By absorbing the constraint into the objective with a Lagrange multiplier $\lambda \in \mathbb{R}^+$, we construct the Lagrangian function $L(\mathbf{r}, \lambda)$:

$$L(\mathbf{r}, \lambda) = \sum_{i=1}^{N} V_i \Big(1 - (1 - P_b)^{S_i} \Big) + \lambda \Big(\sum_{i=1}^{N} \frac{S_i}{r_i} - B \Big)$$

We form a dual function $d(\lambda)$ by minimizing the Lagrangian function for a given λ:

$$\begin{aligned} d(\lambda) &= \min_{\mathbf{r}\in\mathcal{C}} L(\mathbf{r}, \lambda) \\ &= \min_{\mathbf{r}\in\mathcal{C}} \Big\{ \Big[\sum_{i=1}^{N} V_i \Big(1 - (1 - P_b)^{S_i} \Big) + \lambda \frac{S_i}{r_i} \Big] - \lambda B \Big\} \\ &= \Big\{ \sum_{i=1}^{N} \min_{\substack{r_i \in R_j \\ j=1,2,\ldots,K}} \Big[V_i \{ 1 - (1 - P_b)^{S_i} \} + \lambda \frac{S_i}{r_i} \Big] \Big\} - \lambda B \\ &= \Big\{ \sum_{i=1}^{N} \min_{\substack{r_i \in R_j \\ j=1,2,\ldots,K}} L_i(r_i, \lambda) \Big\} - \lambda B \qquad \text{(IV.4)} \end{aligned}$$

where $\mathcal{C}$ is the space of all possible combinations of $r_i, i = 1, 2, ..., N$ selected from $\{R_1, R_2, ..., R_K\}$. This minimization for a given λ can be found by minimizing the sub-Lagrangians $L_i(r_i, \lambda)$ individually; the latter is done by exhaustive search over the discrete set $\{R_1, R_2, ..., R_K\}$. The solution space of the minimization of $L(\mathbf{r}, \lambda)$ is K^N, but because we can minimize sub-Lagrangians individually, we can compute $d(\lambda)$ with only NK evaluations of $L_i(r_i, \lambda)$ and comparisons [83].

We use the subgradient method to search for the best λ in the dual domain. The dual function $d(\lambda)$ is a concave function of λ even when the problem in the primal domain is not convex [83, 84]. Therefore, the optimal λ is found by solving: $\max_{\lambda \in \mathbb{R}^+} d(\lambda)$. Since the dual is a piecewise linear concave function [83], the function may not be differentiable at all points. Nevertheless, subgradients can still be found and used to find the optimal value [83]. It can be shown that the subgradient is a descent direction of the Euclidean distance to the set of the maximum points of the dual function [83]. This property is used in the well-known *subgradient method* for the optimization of a nonsmooth function. The subgradient method is an iterative search algorithm for λ. In each iteration, λ^{k+1} is updated by the subgradient ξ^k of d at λ^k:

$$\lambda^{k+1} = \max(0, \lambda^k + s_k \xi^k / \|\xi^k\|) \tag{IV.5}$$

where s_k is the step size. Based on the derivation in [83], the subgradient ξ^k of the dual function $d(\lambda)$ at λ^k is

$$\xi^k = g(\mathbf{r}^k) - B = \sum_{i=1}^{N} \frac{S_i}{r_i^k} - B \tag{IV.6}$$

where g is the constraint function of the problem, and $\mathbf{r}^k = [r_1^k, r_2^k, ..., r_N^k]$ is the solution to $L(\mathbf{r}, \lambda^k)$.

The step size s_k trades off between the speed of convergence and the variance of the optimized value in each iteration [83]. The complexity of this algorithm is low. In the proposed algorithm, the stepsize is scaled 10 times smaller whenever there is a sign change in subgradient from the previous iteration. When a certain precision of stepsize is achieved, the algorithm terminates. The precision can be chosen differently by context; we used 10^{-18}. By this heuristic method on the change of the stepsize, our method finds the best λ using 82 iterations on average for each optimization. Due to the much lower complexity, we can now consider a larger set of RCPC code rates; the RCPC rates each packet can select from are $\{\frac{8}{9}, \frac{8}{10}, \frac{8}{12}, \frac{8}{14}, \frac{8}{16}, \frac{8}{18}, \frac{8}{20}, \frac{8}{22}, \frac{8}{24}, \frac{8}{26}, \frac{8}{28}, \frac{8}{30}, \frac{8}{32}\}$. Therefore, there are $K = 13$ candidate code rates for our dual search algorithm (**Dual-13**).

Here we further include *"not-sent"* ($r_i = \infty$) and *"uncoded"* ($r_i = 1$) in the RCPC set. The optimization in (IV.1) has the same form. In (IV.1), the P_b corresponding to packets not sent is set to 1; those packets will induce distortion for sure. When we include the options *"not-sent"* and *"uncoded"* to the 13 rates, we now have $K = 15$ and we denote the method **Dual-15**.

Compared to equal error protection (EEP) where there is only one channel rate, we need to signal which channel code rate is used for each packet; this introduces an overhead to each packet. To signal to the decoder which code is used for each packet, we need 4 bits per packet. We assume these bits are collected together and well protected as part of a GOP header. Since the mean packet size is 1917 bits (including both source and channel bits), the 4-bit overhead is negligible.

IV.C Performance of proposed algorithms

In this section, we demonstrate the performance of our end-to-end loss visibility minimization methods. The rate allocation problem is preferably performed across all the packets in a GOP using the rate budget available in a GOP, rather than just considering one frame, so that the number of variables and budget are larger to improve performance.

The video sequence used in our experiment is encoded by H.264/AVC JM Version 12.1 in SIF resolution (352 × 240) with GOP structure IPPP, frame rate 30 fps, and encoding rate 600 *kbps*. We define a packet (a NAL unit) as a horizontal row of macroblocks. Therefore, there are 15 packets in a frame. And a GOP (30 frames) consists of $N = 450$ packets.

The convolutional coder to produce the mother code of the RCPC code

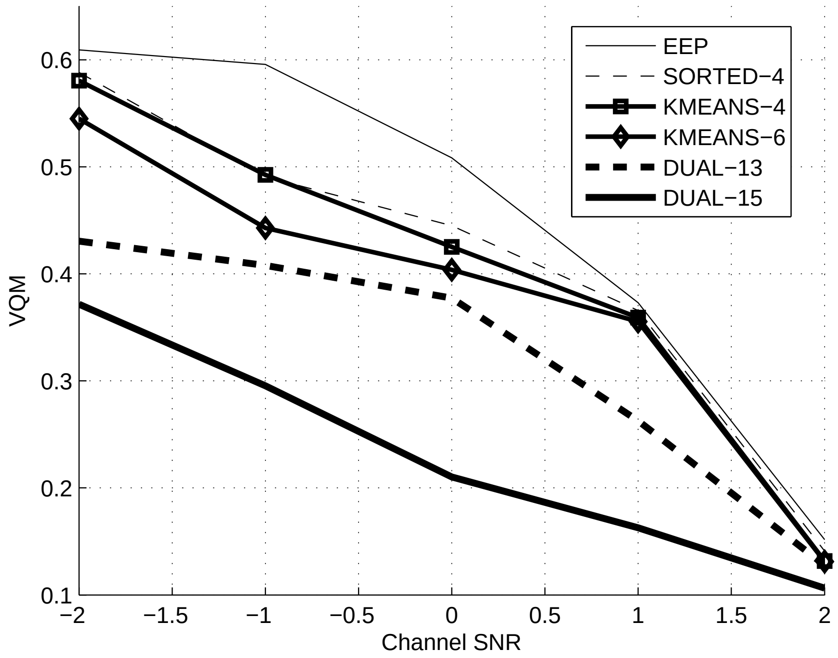


Figure IV.1: The average VQM comparison among six methods over 100 realizations of each AWGN channel.

has rate $\frac{1}{L}$ where $L = 4$, with memory $M = 4$. The puncturing period of the RCPC code is $P = 8$. In the simulation, soft-decision is used for the Viterbi decoder. We simulate an AWGN channel, and find P_b given RCPC code rate and channel SNR. The RCPC rate used by Equal Error Protection **EEP** is $\frac{8}{14}$. The RCPC rates from which our UEP methods can select are $\{\frac{8}{9}, \frac{8}{10}, \frac{8}{12}, \frac{8}{14}, \frac{8}{16}, \frac{8}{18}, \frac{8}{20}, \frac{8}{22}, \frac{8}{24}, \frac{8}{26}, \frac{8}{28}, \frac{8}{30}, \frac{8}{32}\}$, depending on the method we use, as will be discussed below. The bit budget for the optimization problem will be the number of bits used by the EEP in the optimization group. The simulated AWGN channel SNR (defined as E_s/N_o, where E_s is the energy per coded bit, and N_o is the variance of Gaussian noise) ranges from -2 to 2 dB. Instead of using mean square error, the resulting source-decoded videos are evaluated by the full-reference metric VQM (Video Quality Metric) [4]. VQM ranges from 0 (excellent quality) to 1 (poor quality).

Figure IV.1 shows the VQM comparison result among EEP, SORTED-4, KMEANS-4, KMEANS-6, DUAL-13 and DUAL-15. Each data point is obtained by averaging the performance of the corresponding algorithm over 100 realizations of the AWGN channel. We observe that our proposed UEP methods consistently perform better (lower VQM scores) than EEP for all SNRs.

For the method of SORTED-4, the RCPC rates to be considered are $\{\frac{8}{12}, \frac{8}{14}, \frac{8}{16}, \frac{8}{18}\}$. The largest VQM difference between EEP and SORTED-4 is about 0.1, which is considered to be a significant difference in VQM score (see, e.g., [85]).

As discussed earlier, for a GOP, KMEANS-4 is 30 times less complicated than SORTED-4. However, with much less computational complexity, we can see from Figure IV.1 that the VQM result of KMEANS-4 is competitive with and even slightly better than the result by SORTED when the number of RCPC codes is the same ($K = 4$). This is because for SORTED-4, each optimization is performed for the heuristically grouped packets in a GOP, so the bit budget allocation is restricted in each small group of N=15 packets. However, KMEANS-4 can assign the resources to representative packets from the total budget of N=450 packets, and therefore achieve better performance.

Since the complexity is lower for KMEANS-4, we try to improve the performance by offering the algorithm six RCPC code rates $\{\frac{8}{10}, \frac{8}{12}, \frac{8}{14}, \frac{8}{16}, \frac{8}{18}, \frac{8}{20}\}$ to select from. Figure IV.1 shows that at 0 dB, by using 6 RCPC codes, the improvement from EEP increases to 0.1 in VQM score, while at -1 dB, the VQM difference to EEP increases from 0.1 to 0.15. This means that the performance of our visibility-based algorithm makes more difference in the comparison when more RCPC codes are used in the optimization scheme.

Due to the lower complexity of the subgradient method, we can use the full set of the 13 RCPC codes (DUAL-13). We can see from Figure IV.1 that DUAL-13 provides large improvements compared to KMEANS-6 in all channel SNR conditions. There are two reasons for this. First, there are more RCPC code rates from which the optimization process can select. Second, DUAL-13 does not do grouping for packets to perform the optimization; it uses the whole 450 packets in a GOP.

When we add the *"not-sent"* and *"uncoded"* options into the 13 RCPC codes for the optimization (DUAL-15), we get a large improvement even compared to DUAL-13 in every channel condition. Especially, the improvement is about 0.1 in VQM score at SNR=-1 and 1 dB, and 0.2 in VQM score at SNR=0, showing the advantage of not sending and not coding some of the packets, allowing more resources to go to more important packets.

IV.D Experimental results of comparison against existing method

In [57], the ideas of *"not-sent"* and *"uncoded"* are also used for optimal channel code allocation, and the channel codes used are RS codes. The packet importance is measured by D_i, the MSE over a GOP between a compressed/reconstructed video with no packet loss, and one with the ith packet lost. The algorithm does not consider multiple code rates or variable packet size. We consider variable-sized packets and a bit error channel, so packet size needs to be considered in choosing protection, including the options of discarding and no protection. In this section we compare our DUAL-13 and DUAL-15 with their algorithm. To compare, we modified our algorithm by substituting the packet importance measurement from V_i to D_i, and the end-to-end lossy video quality is measured by, instead of VQM, Peak Signal to Noise Ratio (PSNR), calculated by the MSE over all frames between decoded and original videos. Note that because D_i is computed by applying the loss and decoding the *whole* GOP to measure MSE, this is a costly but accurate way of computing D_i, that includes the effect of error propagation.

Although [57] is intended for packet erasure channels and uses RS codes, we made a version using RCPC codes intended for bit error channels. Among the N packets sorted on D_i, the first k_d are discarded, the next k_u are sent uncoded, and the remaining are protected with a single code rate r. We denote this **SortMSE**. SortMSE finds $\{k_d,\ k_u,\ r\}$ that solves:

$$\min_{\{k_d,k_u,r\}} \sum_{\underline{i}=1}^{k_d} D_{\underline{i}} + \sum_{\underline{i}=k_d+1}^{k_d+k_u} D_{\underline{i}} \Bigg\{ 1 - \bigg[(1 - P_b(SNR, 1))^{S_{\underline{i}}} \bigg] \Bigg\}$$

$$+ \sum_{\underline{i}=k_d+k_u+1}^{N} D_{\underline{i}} \Bigg\{ 1 - \bigg[(1 - P_b(SNR, r))^{S_{\underline{i}}} \bigg] \Bigg\}$$

$$\text{subject to} \sum_{\underline{i}=k_d+1}^{k_d+k_u} S_{\underline{i}} + \sum_{\underline{i}=k_d+k_u+1}^{N} \frac{S_{\underline{i}}}{r_{\underline{i}}} \le B$$

$$r \in \{R_1, R_2, ..., R_K\} \qquad \text{(IV.7)}$$

where $\underline{i}$ is the sorted packet index. This problem is solved by the method described in [57]. The proposed Dual method requires evaluating the sub-Lagrangian $L_i(r_i, \lambda)$ $82NK$ times or evaluating the Lagrangian function about $82K$ times. SortMSE, according to [10], requires evaluating the objective function in equation (IV.7) at most $2N$ times. This objective function has complexity comparable to the Lagrangian function. Therefore the complexity comparison between the proposed Dual method and SortMSE is about $82K : 2N$. For the values we used ($K = 13$ RCPC code rates, $N = 450$ packets in the optimization) the complexities are comparable.

For simulation, we used H.264/AVC JM Version 12.1 with SIF resolution (352×240), GOP structure IPPP and IBBP, frame rate 30 fps, and encoding rate 600 *kbps*. The error concealment is frame copy, which is one of the options provided in the JM.12.1 decoder. We define a packet (a NAL unit) as a horizontal row of macroblocks. There are 15 packets in a frame. For each GOP structure, we tested two videos: *Foreman* and *Mother-Daughter*. We optimize over one GOP at a time. As there are 30 frames in a GOP, the number of packets in each optimization is $N = 30 \times 15 = 450$. Again, there are $K = 13$ candidate code rates for our dual search algorithm (**Dual13**), and when we include the options *"not-sent"* and *"uncoded"*, $K = 15$ (denoted **Dual15**).

We simulate an AWGN channel, and find P_b given the RCPC code rate and channel SNR. The RCPC rate used by Equal Error Protection **EEP** is $\frac{8}{14}$, and the budget for the UEP optimization problem is the number of bits used by the EEP in the optimization group. Channel SNR ranges from -2 to 5 dB, corresponding to channel bit error rates from about 10^{-1} to 10^{-3}.

PSNR comparisons among decoded videos for Dual13, Dual15, SortMSE and EEP are performed for *Foreman* and *Mother-Daughter* for IPPP and IBBP GOP structures. All results show similar trends. We present the results for *Foreman* in IPPP in Fig. IV.2(a), *Foreman* in IBBP in Fig. IV.2(b), *Mother-Daughter* in IPPP in Fig. IV.3(a) and *Mother-Daughter* in IBBP in Fig. IV.3(b). We see obvious improvements of Dual15 that allows *"not-sent"* and *"uncoded"* over Dual13 that does not. The advantage of Dual15 takes place in every channel condition, and is more obvious at lower SNR because discarding large or unimportant packets is particularly useful in worse channel conditions. The largest improvements of Dual15 over Dual13 are 3.64 dB, 3.49 dB, 3.59 dB and 3.23 dB for the four comparisons (*Foreman* in IPPP, IBBP, *Mother-Daughter* in IPPP and IBBP). The advantage of discarding also can be observed for SortMSE: for low SNR, SortMSE outperforms Dual13 despite SortMSE allowing only one RCPC code for each optimization group. However for better channels, SortMSE is worse than Dual13 because in better channels, packets are less likely to be discarded, so the discarding option of SortMSE can not compensate for its having only one code rate. At better SNRs, SortMSE performs slightly worse than EEP. One might think that SortMSE should never do worse than EEP. EEP only assigns one code rate to all packets, whereas SortMSE chooses $\{k_d,\ k_u,\ r\}$ and should be more flexible. However, by basing the importance on distortion with no consideration of packet size, SortMSE may discard tiny packets that would have cost little to retain, or may heavily protect large packets that are costly to retain, and may do worse than EEP. However in worse channels, the flexibility of being able to discard packets compensates for this disadvantage. Dual15 that features both packet discarding and flexible rate allocation for each packet performs the best for every channel

SNR.

The results so far use rate $\frac{8}{14}$ for the EEP. The total number of bits after channel coding by this EEP is the constraint for the optimization. It is possible that $\frac{8}{14}$ is particularly unsuitable for some channel SNRs. To show that Dual15 performs better than EEP for our entire set of possible EEP rates, we separately channel encode the same pre-encoded video sources using each different EEP (excluding the possibility of discarding everything). This gives rise to 14 different bit constraint totals. For each, we ran Dual15. The average improvement of Dual15 over the corresponding EEP is $\{13.2, 12.3, 9.8, 7.2, 5.6, 4.3, 3.5, 2.4, 1.7, 0.9, 0.5, 0.3, 0.09, 0\}$ dB. Dual15 outperforms EEP in all cases, even with a tiny bit budget (EEP = “uncoded”), because Dual15 can discard some packets to free up bits for protecting important packets. The advantage of Dual15 decreases as total bits increase, until finally when each packet is equally protected by the strongest channel code, the improvement of Dual15 over EEP vanishes, because at that high total bit rate Dual15 and EEP both can afford to equally protect every packet with the strongest rate. In summary, Dual15 outperforms all the EEP values except for the extreme cases of maximal protection and total discarding where they are equal.

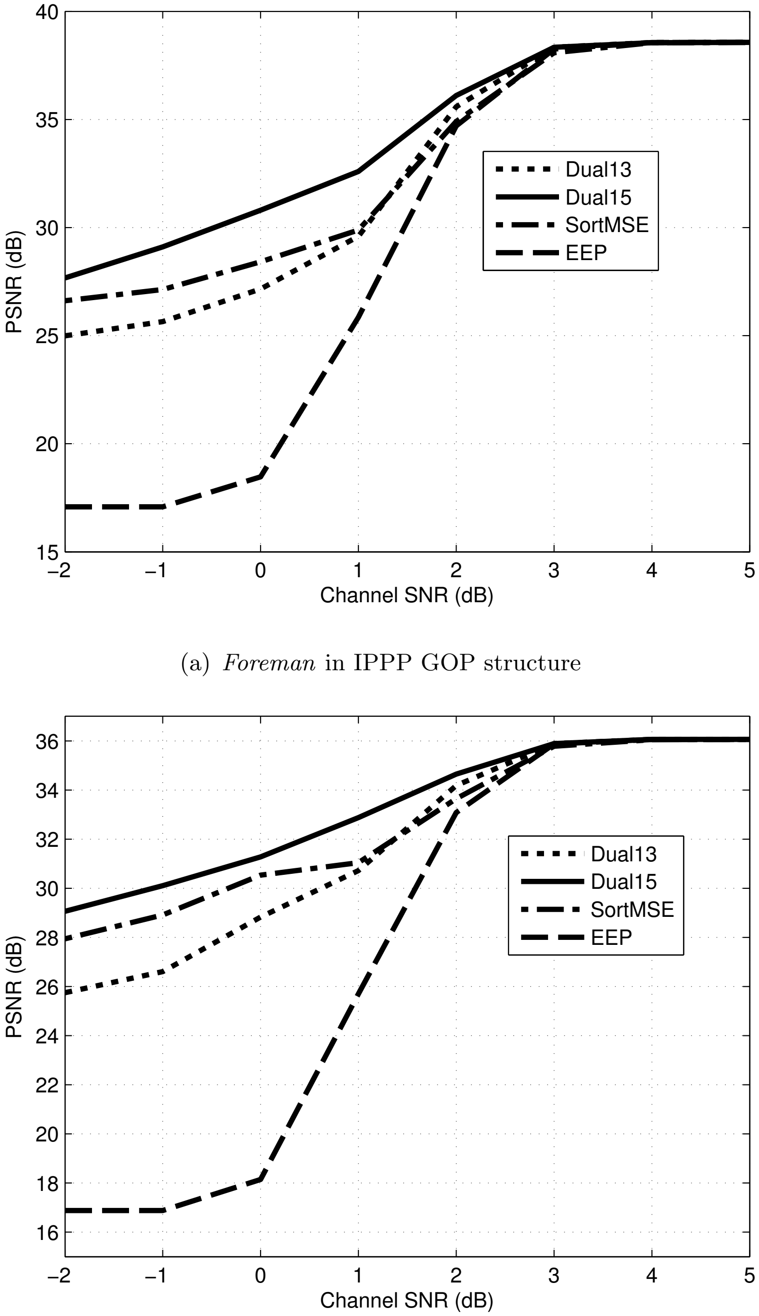
IV.E Conclusion

In conclusion, we use the branch and bound method to solve the channel rate RCPC allocation problem to reduce the end-to-end packet loss visibility over an AWGN channel. The result shows that our method consistently achieves better end-to-end video quality in different channel SNRs than Equal Error Protection. Then we proposed a much less complicated algorithm using K-means clustering that performs better since no heuristic grouping is used. And because of the lower complexity, the code rate set can be enlarged and better visual performance is achieved. We further develop an algorithm that searches in the dual domain by the subgradient method for the optimal channel code rate for each packet with different packet size and different importance. The algorithm is of low complexity. We exploit the options of not coding and not sending the packets. The algorithm improves the performance considerably. We also compare this algorithm with a simpler UEP version that considers only options of discarding, not coding, and a single level of protection. For all channel conditions, video clips and GOP structures tested, our algorithm significantly outperforms it, as well as the equal error protection.

IV.F Acknowledgements

This work was supported in part by the National Science Foundation, the Center for Wireless Communications at UCSD, and the UC Discovery Grant program.

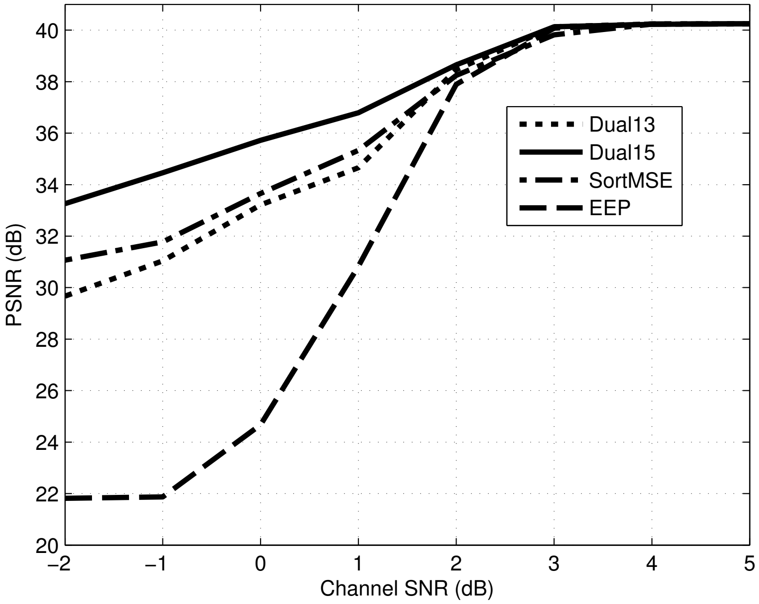
Chapter IV of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin and P. Cosman, “Optimal RCPC channel rate allocation in AWGN channel for perceptual video quality using integer programming”, *QoMEX 2009, First International Workshop on Quality of Multimedia Experience*, July, 2009, and in T.-L. Lin and P. Cosman, “Perceptual Video Quality Optimization in AWGN Channel Using Low Complexity Channel Code Rate Allocation”, *43rd Asilomar Conference on Signals, Systems and Computers*, 2009, and in T.-L. Lin and P. Cosman, “Efficient optimal RCPC code rate allocation with packet discarding for pre-encoded compressed video”, *IEEE Signal Processing Letters*, vol. 17, issue 5, pp. 505-508, 2010. I was the primary author of these three papers. Coauthor Prof. Cosman directed and supervised the research which forms the basis for Chapter IV.



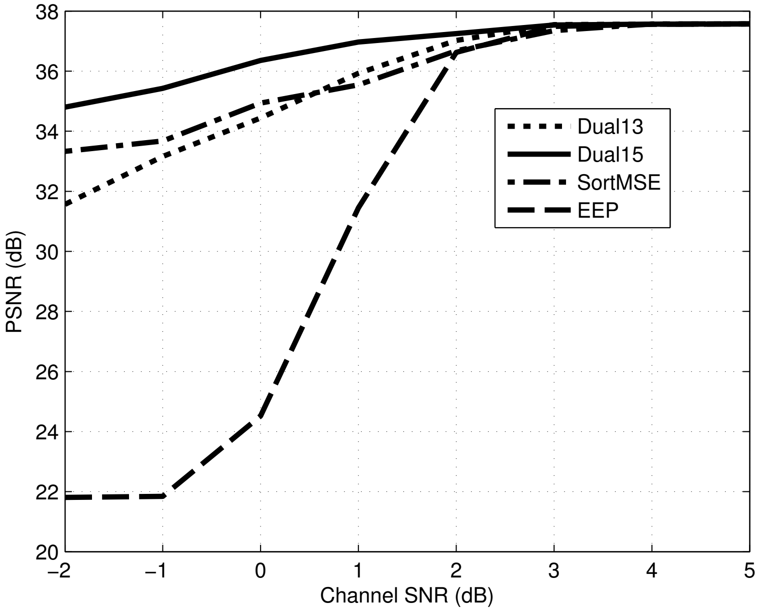
(a) *Foreman* in IPPP GOP structure

(b) *Foreman* in IBBP GOP structure

Figure IV.2: *Foreman*. Average PSNR of decoded video vs. channel SNR. Comparison among Dual13, Dual15, SortMSE and EEP over 100 realizations of each AWGN channel.



(a) *Mother-Daughter* in IPPP GOP structure



(b) *Mother-Daughter* in IBBP GOP structure

Figure IV.3: *Mother-Daughter*. Average PSNR of decoded video vs. channel SNR. Comparison among Dual13, Dual15, SortMSE and EEP over 100 realizations of each AWGN channel.

V

Network-based packet loss visibility model

Since different video packets have different impact on visual quality when dropped, it is important for an intermediate router to estimate the visual importance of each packet to know which ones to drop during congestion. The previous chapters focused on encoder-based models which are used to estimate the visual importance of each packet. In Chapter III, the packet is embedded with a priority bit at the encoder so that the router could perform smart dropping during congestion. The model requires factors such as Initial MSE, type of camera motion, information on the reference frame and on scene cuts. This is applicable at the encoder where the reference frame is available, and where the computational capability is high.

In contrast, the current chapter focuses on a network-based model where the complexity must be limited, and in any case, reference frames are not necessarily available because packets may be out of order or because there are multiple streams and the network node cannot afford to decode and reconstruct them. This model is intended to be a good tool to evaluate the perceptual importance of incoming packets that do not have prioritization bits.

Another goal is to explore the difference between SDTV and HDTV packet visibility models. Subjective results in [58] showed that displays should guarantee a large screen with high contrast to achieve the higher expectation for watching HDTV than for watching SDTV. The work in [59] concluded that people prefer SDTV with high quality over HDTV with low quality. These works comparing SDTV and HDTV are not concerned with packet loss visibility. One related paper is [60], which studied region of interest (ROI) determination for SDTV and HDTV. The study showed that the ROI of a video is identical for both SDTV and HDTV. Also, losses occurring in the top and the bottom regions of the picture were not generally in the ROI.

This chapter is organized as follows: In Section V.A, the subjective tests are described. In Section V.B, we discuss self-contained factors that relate to packet loss visibility, and the models based on these factors. Section V.C presents results and discussion.

V.A Subjective experiments

The video encoder is H.264 JM9.3. Encoder settings (Table V.1) adhere to ITU and DSL Forum Recommendations [86, 87]. Each Network Abstraction Layer (NAL) packet contains a horizontal row of MBs in a frame. There are 30 packets per SD frame, and 68 per HD frame. The raw video sources are in HD format, and the SD versions are obtained by downscaling the HD videos by bicubic interpolation. Nine videos with widely varying motion and texture characteristics are concatenated into a 20-minute sequence.

The decoder is FFMPEG [65] due to its high efficiency and wide use in industry. For error concealment for non-whole-frame losses, the FFMPEG decoder begins by making a guess, for each lost macroblock, of whether it is more likely to have been intra coded or inter coded. For example, in P and B frames, the algorithm looks at the coding mode of some or all of the macroblocks that are not lost, and if more than half of those have a coding mode which is intra, then the algorithm will guess that all the lost macroblocks in the frame were coded intra. Once the guess has been made of coding mode for each lost macroblock, the algorithm uses two different approaches. For the macroblocks which are guessed to be intra coded, for each 8×8 block in each MB, ffmpeg does a process called FillDC, which looks at the four directions surrounding the block (top, bottom, left, right) to find uncorrupted blocks. It then finds the pixel average of each uncorrupted neighboring block. Finally, it takes a weighted average (weighted according to distance) of the uncorrupted averaged blocks, and the result is the block that is used for concealment. For the macroblocks which are guessed to be inter coded, the algorithm estimates the forward and backward motion vectors by scaling, in display distance, the co-located future and past (in display order) motion vectors in the buffer. The obtained motion vectors are used to perform bi-directional motion estimation to conceal the lost MB. The whole frame losses are concealed by temporal interpolation of closest past and future (in display order) reference frames.

Each subject watches a lossy HD video and the corresponding SD version, 40 minutes in total. The experiment takes one hour, which includes an introductory session and a break. When viewers see a glitch, they press the space bar. To allow observers enough time to respond to each individual loss, only one packet loss occurs for every 4-sec interval. The loss occurs in the first 3 seconds, and the fourth second allows any error propagation to terminate. During the 40 minutes of video, there are 600 packet loss data points obtained from a subject. These losses are divided equally among I frames, P frames and B frames. There are three different loss realizations; each of the three 40-minute lossy video pairs is watched by 10 people. The ground truth packet loss visibility for a specific packet can be obtained as the number of people who see the loss artifact divided by 10. With three loss realizations, each evaluated by 10 people, we have ground truth visibility for 600×3=1800 packets (900 for SD, 900 for HD resolution).

Table V.1: Summary of the subjective experiment setup for SD and HD videos. H is the height of the video.

	SD	HD
Resolution	720 × 480	1920 × 1080
Bitrate	2.1 Mbps	10 Mbps
H.264 Profile	Main profile Level 3	Main profile Level 4
Viewing Distance	6H	3H
Frame rate	30 fps	
GOP	IBBPBBPBBPBBPBB 15/3	

V.B Features and model building

In this section, we first introduce candidate factors associated with a packet. Next, we build models using these parameters to predict, for each packet, the packet loss visibility results of our subjective experiment.

V.B.1 Important features

Content dependent factors depend on the actual video content at the location of the loss. The ones we use all involve taking a mean, maximum, or variance computed over all macroblocks in the packet. **MeanRSENGY** is the mean residual energy after motion compensation. **MaxRSENGY** denotes the maximal residual energy after motion compensation. Following the way these factors were used in Chapter II, we used the above two terms after logarithm because they were shown to be more correlated with packet loss visibility (we add 10^{-7} before taking the log to avoid a log of zero problem). **MeanMotX** and **MeanMotY** are the mean motion vectors in the x and y directions. **MaxMotX** and **MaxMotY** are the maximal motion vectors. **VarMotX** and **VarMotY** are the variances of the motion vectors. **MotM** is $\sqrt{MeanMotX^2 + MeanMotY^2}$. To compute the factors related to phase of motion vectors, we only consider macroblocks with non-zero motion, for which the phase is well defined. **MeanMotA** is the mean phase. **MaxMotA** is the maximal phase. **MaxInterparts** is the maximal number of inter macroblock partitions in the packet.

Content independent factors depend on, for example, the spatial location or frame type of the loss, but do not depend on the actual video content at the location of the loss. **TMDR** is the maximum number of frames to which the error from this packet loss can propagate. TMDR=1 for non-reference frames. For reference frames, TMDR depends on the distance to the next I frame. **Height** is the spatial location where the loss occurs; the top slice in a frame has Height=1, and the bottom slice in a frame has Height=N, where N is the number of packets in a frame (30 for SDTV and 68 for HDTV). Most of the factors mentioned above have a monotonically increasing (or decreasing) relationship with the average packet loss visibility. However, this is not the case for Height. The plots of average packet loss visibility versus Height are in Fig.V.1. Although the data are noisy, we see the trend that average packet loss visibility is highest near the middle of the screen, and decreases as we move to the top or bottom. This is difficult to capture by a linear relation, therefore, we create **DevFromCenter** = abs(Height-floor(N/2)) to indicate how far away the loss occurs from the vertical center of the frame.

In addition to these content independent and content dependent factors, we also consider the interactions between factors in one category and factors in the other, as well as between factors within the content independent category.

The motion information mentioned above is estimated by the network node where reference frames are not available. 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 from the previous co-located MB. Within a packet, 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 packet level, and can be implemented

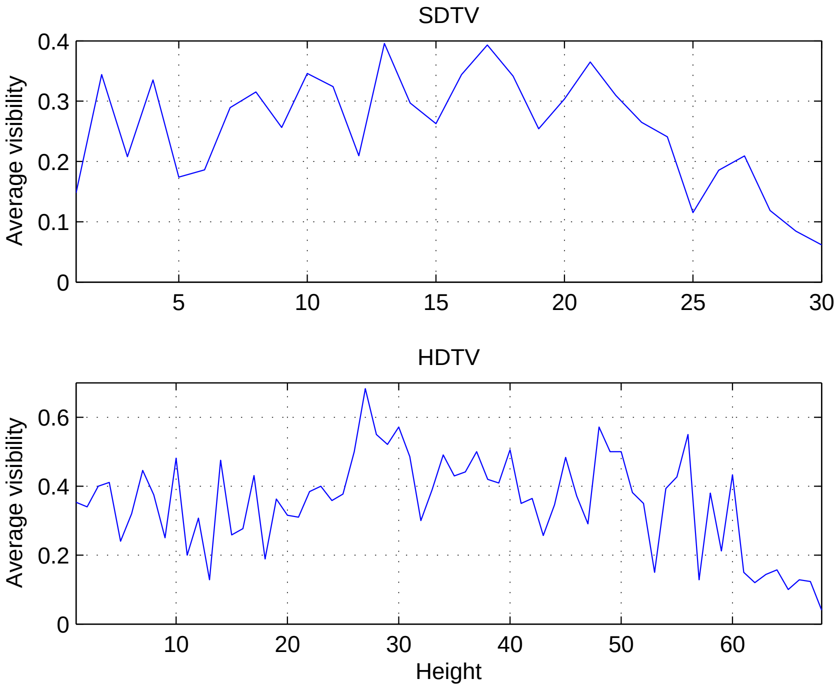


Figure V.1: Average packet loss visibility versus Height

at a network node.

V.B.2 Modeling process

We choose a generalized linear model (GLM) with the logit function as link function, since it can predict a probability parameter in a binomial distribution. We want to know the probability that a packet loss artifact will be observed when the packet is lost. The background of GLMs was introduced in Chapter I.

In Chapter II, the GLM development treats data points equally, no matter how far they are from the regression line. In this Chapter, we give unequal treatment to data points to suppress outliers by minimizing the M-estimator. The detailed discussion is presented in Chapter I.

We develop GLM models for both SD and HD resolution videos. The best factors chosen for them and their corresponding coefficients are listed in Tables V.2

and V.3. Figure V.2 shows the decrease of the M-estimator as additional factors are incorporated in the SD and HD models.

V.C Results and discussion

From Figure V.2, we see the best M-estimator value is 0.1096 for the SDTV model and 0.1201 for the HDTV model. If we compare against an encoder-based model which uses initial MSE, requiring the reference frame and frame reconstruction, as a factor, the encoder-based models perform better as expected; the minimum achievable M-estimators are 0.1067 for SDTV and 0.1172 for HDTV, as shown in Figure V.2. However, the performance difference is slight; the network-based model performs almost as well as the encoder-based model, but the former is suitable for a router as it uses no information from reference packets or pixel domain processing.

We can not properly interpret the model by the sign of the coefficients in Tables V.2 and V.3 if the factors correlate with each other [88], however the order in which factors are added to the model provides an indication of their importance. The most important factors in both SDTV and HDTV relate to TMDR, indicating that error propagation duration dominates the packet loss visibility regardless of resolution. However the spatial location of the loss affects the visibility differently between models. In Fig.V.1, the maximum average loss visibility is 0.3957 at Height=13 for SDTV, and 0.6833 at Height=27 for HDTV; they are both near the middle slice. The minimum average loss visibility is 0.0615 at Height=30 for SDTV, and 0.0400 at Height=68 for HDTV; they are both at the bottom slice. Packet losses in the center are more visible than those at the bottom. What is more, given that the average packet loss visibility for all losses in SDTV is lower than that in HDTV (0.2565 and 0.3506), it is surprising that the average loss visibility of the bottom packet in HDTV is lower than that in SDTV, and the ratios of maximum loss visibility to minimum loss visibility are 6.4341 and 17.0825

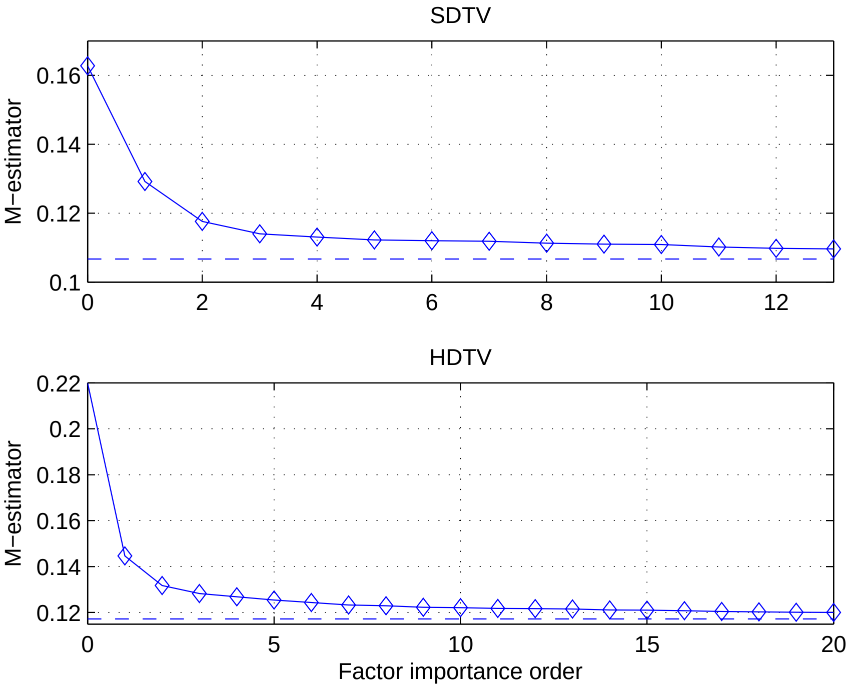


Figure V.2: M-estimator value decreases as important factors are included in the SDTV and HDTV models. Numbers on x-axis denote the index in factor order shown in Tables V.2 and V.3. The dashed horizontal line denotes the minimum M-estimator value of the SDTV and HDTV encoder-based models.

for SDTV and HDTV respectively. In the viewing conditions of Recommendations [86, 87], HD requires a larger viewing angle. The viewing angles are (vertical, horizontal)=$(9.52°, 14.25°)$ for SDTV, and $(18.92°, 33°)$ for HDTV. Therefore, a viewer who watches HDTV may not fully realize what happens in the edge area of a frame. Prior research [60] found that losses occurring in the top and the bottom regions of the picture were not generally in the region-of-interest. We would add to this that, for HDTV, losses occurring at the top and bottom are less likely to be noticed not only because they are not in the ROI but also because of the larger viewing angle.

Table V.2: Table of factors in the order of importance for SD GLM model. The × symbol means multiplication. Bolded factors relate to spatial location.

Order	Factors	Coefficients
α	1	-2.6407
1	TMDR×MaxMotA	-4.7591e-3
2	**DevFromCenter**×MaxMotA	2.2996e-2
3	**Height**×MeanMotA	-8.8462e-4
4	TMDR×log(MeanRSENGY $+10^{-7}$)	3.5954e-3
5	TMDR×MeanMotY	-1.6431e-2
6	**DevFromCenter**×TMDR	-1.0164e-2
7	**DevFromCenter**×MeanMotY	5.3172e-3
8	TMDR	2.3680e-1
9	TMDR×MaxInterparts	-5.6283e-3
10	TMDR×MotM	4.9349e-3
11	**Height**×**DevFromCenter**	-3.1830e-3
12	**Height**×MaxInterparts	2.1661e-3
13	TMDR×VarMotY	5.1232e-4

V.D Conclusion

We propose self-contained packet loss visibility models for SDTV and HDTV. These network-based models perform only slightly less well than the much more complicated non-self-contained models that could be implemented only at the encoder. The proposed models allow a network node to efficiently evaluate the visual importance of packets just by information contained in each packet. No reference information or frame reconstruction is required for the prediction factors. This model can be useful to evaluate packets in the network in case of congestion. The study found that packet loss is more visible in HDTV than in SDTV. And due to the wider viewing angle for HDTV, the spatial location of the packet loss in HDTV matters more than in SDTV. For both SDTV and HDTV models, the temporal duration of the error propagation is a very important factor for a packet to be visible.

Table V.3: Table of factors in the order of importance for HD GLM model. The × symbol means multiplication. Bolded factors relate to spatial location.

Order	Factors	Coefficients
α	1	-3.0413
1	TMDR×log(MaxRSENGY $+10^{-7}$)	9.1743e-3
2	**Height**×**DevFromCenter**	-2.1129e-3
3	**Height**×TMDR	3.4239e-4
4	TMDR×MaxMotA	6.0561e-2
5	**Height**×MotM	9.9631e-4
6	**Height**	3.2186e-2
7	**DevFromCenter**×MeanMotY	1.3397e-3
8	**Height**×VarMotX	-2.0544e-5
9	TMDR×VarMotX	3.8690e-4
10	TMDR×MeanMotX	3.3589e-3
11	**DevFromCenter**×TMDR	-4.7789e-3
12	log(MaxRSENGY $+10^{-7}$)	-6.5376e-2
13	**DevFromCenter**	7.6811e-2
14	**Height**×MaxInterparts	7.9892e-4
15	**DevFromCenter**×MaxInterparts	-9.3612e-4
16	**DevFromCenter**×MaxMotY	-6.7759e-4
17	**DevFromCenter**× log(MeanRSENGY $+10^{-7}$)	3.9123e-3
18	TMDR×MeanMotY	2.1333e-3
19	VarMotY	2.3235e-4
20	TMDR×log(MeanRSENGY $+10^{-7}$)	3.1425e-3

V.E Acknowledgements

This work was supported by Futurewei Technologies, Inc. and by the Center for Wireless Communications at UCSD.

Chapter V of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin and P. Cosman, “Network-based packet loss visibility model for SDTV and HDTV for H.264 videos”, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2010.* Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter V.

VI

Packet dropping using a network-based loss visibility model

In past work, dropping decisions made by an intelligent router depend on MSE induced by a packet loss. Especially in [36], an intermediate router with an optimization algorithm drops packets in a congested network from different streams to minimize the sum of cumulative MSE. The packets must have embedded information about the associated induced MSE. Also, the optimization process is too complex for most current routers. Furthermore, MSE does not correlate well with human perception [11]. The works [61, 62] are for No-Reference network monitoring, which can compute estimated video quality for a given packet loss pattern using only video bitstream information. However, they give an overall quality score for the sequence and do not tell us how to best drop packets to minimize the video quality degradation during network congestion.

In Chapters II and III, the visual importance of each packet is evaluated in the encoder by an *encoder-based* packet loss visibility model. Every piece of 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. One limitation of Chapter III is that the priority score needs to be determined at the encoder and added as one bit to the packet header.

In this chapter, we 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 Chapter V which requires information only within one packet and no reference frame information. This is desirable because packets may be out of order or because there may be multiple streams and the network node cannot afford to decode and reconstruct them. The parameter extraction process can be made very efficient since it does not involve motion compensation (requiring reference frame) and frame reconstruction. Second, we devise a packet dropping method for widely varying packet loss rates including high rates. The algorithm drops the least visible *frames*, which incurs fewer blocky artifacts compared to dropping on a *packet* basis. This method is shown to be better than a method used by industry which drops B packets, for different levels of packet loss rate in terms of VQM scores.

This chapter is organized as follows. Section VI.A discusses the proposed multiple packet loss algorithm using measurements obtained by the network-based packet loss visibility model. Simulation results are in Section VI.B, and Section VI.C concludes the chapter.

VI.A Multiple packet loss algorithm

In Chapter III, the packet dropping policy used the fact that each incoming packet to the router has a 1-bit prioritization bit to signal whether this packet is estimated to be of low or high packet loss visibility. The loss visibility estimation is done at the encoder. In this chapter, we assume no prioritization bit and use the network-based model to estimate the visual importance of each incoming packet. This model predicts the packet loss visibility of each packet based only on the information within one packet (NAL in H.264). Also the model does not need information from the pixel domain, and therefore the tasks of motion compensation, deblocking filtering and frame reconstruction are not necessary. Therefore the complexity to obtain the factors for prediction is much lower than a full decoder and is more realistic for implementation in a router.

We define bit reduction rate (BRR) as the percentage of bits that need to be dropped of the buffered packets to alleviate the congestion. Given the packet loss visibility scores, during network congestion, a router can straightforwardly drop packets with least estimated visibility until the required bit reduction rate is achieved and the congestion is relieved. We denote this method **Vis-Pkt**. An intelligent dropping method that is implemented in a video-aware digital subscriber line access multiplexer (DSLAM) is discussed in [89]. 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. We denote this method **B-Pkt**.

In [63], subjective test results showed that in general, frame freezes are less noticeable than blockiness resulting from random packet loss. That is, when one portion of a frame is lost, and another part is not lost, the concealment may cause a spatial misalignment between objects/background in the intact portion of the frame and objects/background in the lost and concealed portion, as shown in Figure VI.1. This spatial misalignment in a frame often draws more attention from viewers than does a frame freeze, which has no spatial misalignment problem. Frame freeze can be produced by whole frame loss when the decoder conceals the lost frame by “frame copy”. This motivated us to consider dropping packets on a *frame* basis, instead of dropping on a *packet* basis, as done in Vis-Pkt and B-Pkt. We design an algorithm, denoted **Vis-Frame-Pkt**. The summed visibility over all packets in a frame is calculated for each frame. We drop the N least visible frames. N is chosen such that the total number of bits in the packets comprising these frames is under the total number of bits of BRR, but dropping the (N+1) least

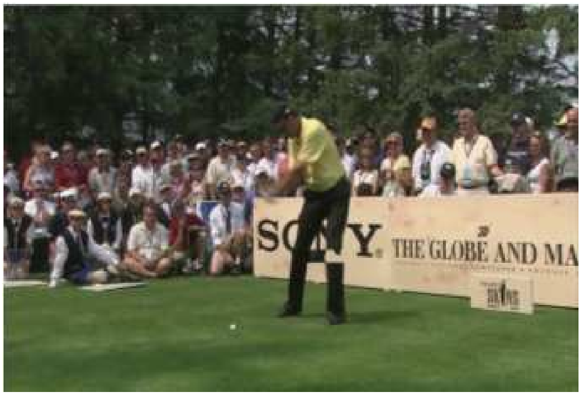


Figure VI.1: An example of spatial misalignment

visible frames would put the total over BRR. Then we drop packets on a packet basis to reach the required number of bits of BRR. We design a similar algorithm, denoted **B-Frame-Pkt**, which randomly drops B packets on a frame basis, and when dropping the next B frame would mean dropping more than BRR bits, it switches to dropping on a packet basis to reach the BRR bits.

Since the size of a packet is much less than that of a frame, these methods that switch to dropping on a packet basis try to meet very closely the goal of number of bits to be dropped. However, spatial misalignment is introduced by this approach since some packets are dropped on a packet basis rather than on a frame basis. Another approach is to drop the (N+1) least visible frames all the way until the goal of number of bits to drop has been reached, even though this means that, in general, the goal will be exceeded by perhaps a large number of bits due to the granularity of dropping whole frames. We denote this method **Vis-Frame**. And the counterpart of this in dropping B packets is denoted **B-Frame**, which randomly drops B frames until the requirement is reached.

VI.B Experimental results

In this section, we compare the six methods for different videos and different levels of BRR. All the methods which relate to dropping B packets are implemented by randomly dropping B packets/frames in the buffer, and when running out of B packets/frames to be dropped, P packets/frames are dropped randomly. The performance is evaluated by averaging 50 random realizations.

The video encoder is H.264 JM9.3. The decoder used is FFMPEG [65]. The resolution is SDTV. The tested videos are encoded at 2.5Mbps, 30 fps using Main profile Level 3. The GOP structure is IBBP (15 frames). Each NAL packet comprises one horizontal row of macroblocks. Therefore we have 30 packets in a frame. The error concealment strategy in FFMPEG is described in Section V.A.

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 decoding and error concealment are run, and then the Video Quality Metric (VQM) [4] is calculated to obtain the video quality score for this lossy GOP. It ranges from 0 (excellent quality) to 1 (poorest possible quality).

Two videos are tested in the simulation: *golf* is of slow movement, and *soccer* is of high motion and fast panning. The simulated BRRs are 0.5%, 5%, 10% 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 two videos are subsets of the nine videos used for the subjective experiments, however, the losses injected for this section are very different from those in the subjective experiments because the latter used isolated slice losses.

Figure VI.2 shows the VQM performance versus GOP index for the six dropping methods for BRR = 0.5%, 5%, 10% and 20% for the *golf* video. Figure VI.3 shows VQM score averaged over GOPs versus BRR for the six packet dropping policies. From the figures, we observe the general trend that the -Frame-Pkt method is better than the -Pkt method for both the Vis and B methods. This means that dropping packets on a frame basis helps the video quality. We also observe that the -Frame method is better than the -Frame-Pkt method in general.

This means even though -Frame drops more bits than BRR needs, the perceptual quality is improved due to having no spatial misalignment (blockiness) problem.

Note that the y-axis scale changes steadily as one looks at Figure VI.2(a), then VI.2(b), VI.2(c) and VI.2(d). This is because higher BRR makes the quality generally worse, so VQM scores become higher and have more variance. Note also that the y-axis scale for Figure VI.3(a) is not the same as for Figure VI.3(b). Because soccer is a high motion video, and golf is a low motion video, losses are less concealable for soccer. Therefore, for a given BRR, the scores for soccer are worse (higher) than they are for golf. So, the average improvement in VQM score for the best dropping approach compared to the worst one is very much larger for soccer than for golf. While the average improvement in VQM score (provided by the best dropping method) shown in Figure VI.3(a) for the golf sequence is very small, and so might be considered perceptually not noticeable, if one looks at the VQM scores versus GOP index shown in Figure VI.2, many of the individual GOPs have substantial VQM improvements, which would be perceptually noticeable.

For comparisons between B- methods and Vis- methods, when the BRR is very low (0.5%), Vis-Pkt is better than B-Pkt. However, when the BRR increases, Vis-Pkt is not better than B-Pkt. It may be that Vis-Pkt does better at low dropping rates because the visibility model is developed from videos with isolated losses where the evaluated packets have their intact reference frames, which is not the case for much higher packet loss rate. However when we perform whole frame drop (Vis-Frame-Pkt or Vis-Frame methods), we can see from Figure VI.3 that for all BRRs, both Vis-Frame and Vis-Frame-Pkt are better than B-Pkt, B-Frame-Pkt, and B-Frame. Lastly, we can observe that Vis-Frame is always better than Vis-Frame-Pkt except at the lowest loss rates. This makes sense because for low loss rates, the fact that Vis-Frame exceeds the target BRR in order to maintain dropping of whole frames incurs a more severe penalty percentagewise in bits dropped.

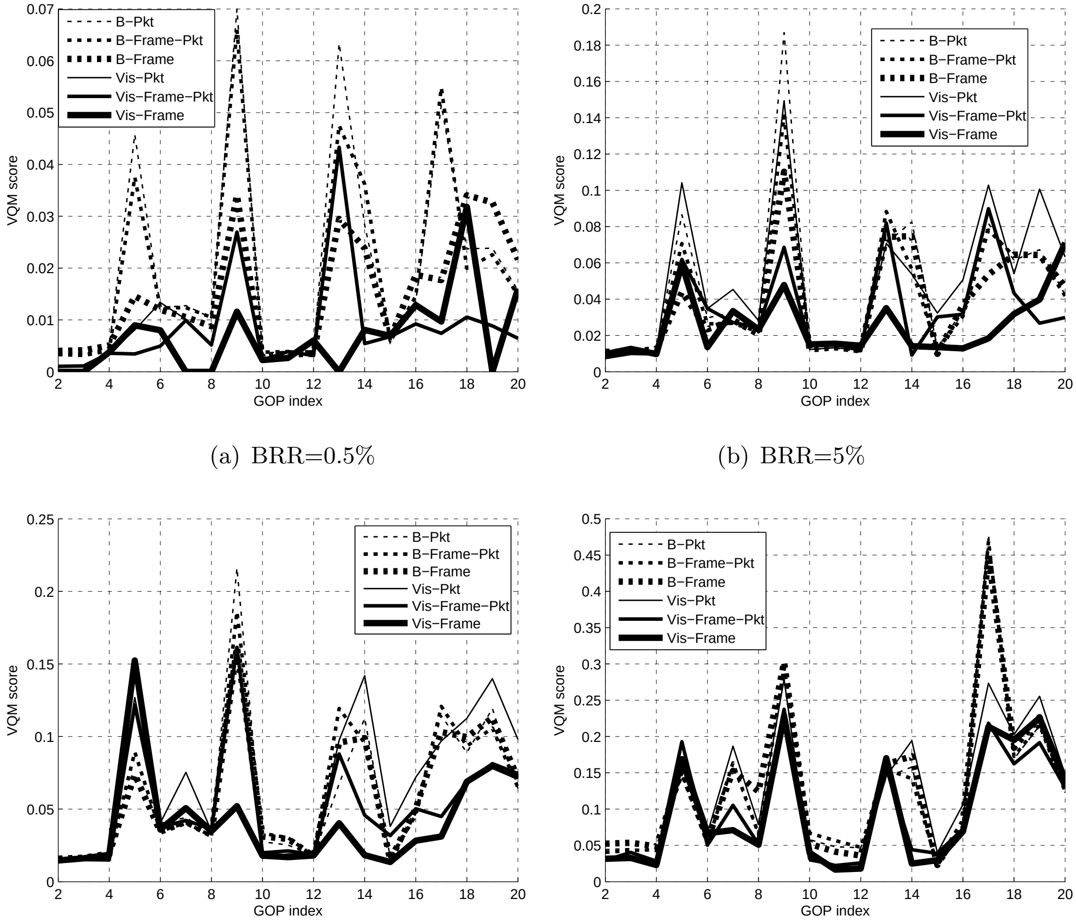
VI.C Conclusion

We used a network-based packet loss visibility model to measure the visual importance of packets incoming to a router. The estimated visibility scores are then used by the router to perform intelligent packet dropping. For a very wide variety of bit reduction rates, the performance of the proposed algorithm outperforms both a visibility-based algorithm that drops packets on a packet basis, and an algorithm that drops B packets or B frames such as one currently implemented in a video-aware digital subscriber line access multiplexer. The contributions of this chapter are (a) We showed that dropping whole frames and concealing by simple frame interpolation produces better quality video than dropping on a packet (slice) basis, (b) We showed that the visual advantage of dropping whole frames is sufficiently large that, except for low dropping rates, it pays to drop whole frames even when that means exceeding the target bit reduction rate for the GOP, (c) A simple visibility model that can be implemented inside the network provides a better basis for choosing frames to drop than just targeting B frames for dropping.

VI.D Acknowledgements

This work was supported by Futurewei Technologies, Inc. and by the Center for Wireless Communications at UCSD.

Chapter VI of this dissertation, in part, is a partial reprint of the material as it appears in T.-L. Lin, J. Shin and P. Cosman, “Packet dropping for widely varying bit reduction rates using a network-based packet loss visibility model”, *IEEE Data Compression Conference (DCC), 2010.* Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter VI. Co-author J. Shin also contributed to the ideas in this work.



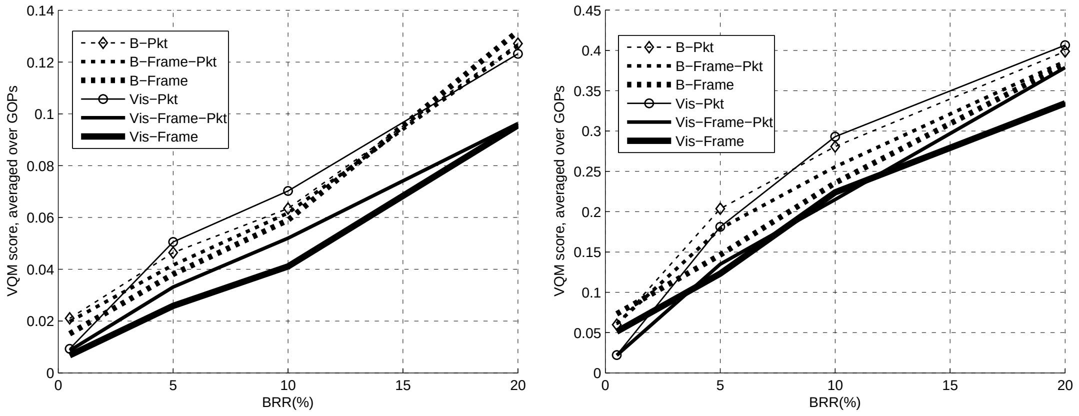
(a) BRR=0.5%

(b) BRR=5%

(c) BRR=10%

(d) BRR=20%

Figure VI.2: VQM performance vs. GOP index for the six packet dropping policies for SDTV *golf* for BRR = (a) 0.5% (b) 5% (c) 10% and (d) 20%. Lower VQM scores correspond to higher quality.



(a) *golf* (b) *soccer*

Figure VI.3: Average VQM score over GOPs vs. BRR for the six packet dropping policies for SDTV videos (a) *golf* and (b) *soccer*. Lower VQM scores correspond to higher quality.

VII

Whole frame packet loss visibility

The packet loss visibility modeling in previous chapters was designed for packets that are just slices (defined to be one horizontal row of macroblocks) of a frame. For these types of packet 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 copy [63]. In Chapter VI, under the same dropping target, we dropped packets on a slice basis, and on a frame basis. We found that the frame-level temporal interpolation artifact is better than the slice-level spatial misalignment artifact using VQM scores.

Nevertheless, which whole frame to be dropped in Chapter VI was estimated by the visibility model for single-slice packets. That is, the visibility score for the frame was taken to be simply 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 involving slice loss data. Therefore, to obtain more meaningful scores for frame losses, in this chapter, we conduct a subjective experiment to concentrate on the subjective results for whole frame loss, and build a direct model for whole frame loss. Two common concealment methods for whole frame losses are frame copy and temporal frame interpolation. In this experiment, we simulate frame copy by the frame copy error concealment in the JM standard decoder [64], and frame interpolation by FFMPEG [65]; these two decoders are popular in research and industry. In this chapter we analyze the experimental data, and model the whole frame packet loss visibility based on information associated with the lost frames.

We hope to build a model that is suitable for router operation so that in the case of network congestion, the router is able to decide based on our model which frame or frames to drop to relieve the congestion while maintaining good video quality. Therefore, as in Chapter V, we consider factors associated with the frame considered for dropping to be self-contained, meaning that the computation of the factors does not need other (reference) packets. 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 we want the complexity of the factor extraction process to be low. Therefore we 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 Chapter II.

Perceptual quality of frame losses is discussed in [66]. The work studies different whole frame loss type as a function of frame loss burst length and frame loss burst distribution. The authors conclude that the visibility of frame dropping is dependent on content, loss duration and motion. Later, in [67], they built an assessment model for subjective video quality as a function of frame loss burst and frame loss burst distribution. However, the quantities are computed in the pixel domain and require the original video. And the model aims to evaluate the overall quality of a lossy video, and does not indicate the visual importance of a specific frame.

This chapter is structured as follows: in Section VII.A, the setup of the subjective experiment is introduced. Section VII.B covers the analysis of data, and Section VII.C introduces the whole frame loss modeling process and feature selection. Section VII.D concludes the chapter.

VII.A 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 configured in the same way as in Chapter V, except that here we only consider SDTV videos. The decoders we considered are the JM 9.3 standard decoder [64] which produces frame copy artifacts, and FFMPEG [65] 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 frame (in coding order). 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 frames. These two decoders are widely used in academia and industry.

In this experiment, we concentrate on B frames. We introduce whole frame losses once every 4 seconds to allow observers enough time to respond to each individual loss. The losses occur in the first 3 seconds of each 4-second interval. Among these intervals, we inject evenly single or dual whole frame losses in a GOP; we want to understand the visual response to isolated whole frame losses and any interaction between nearby whole frame losses. In this work, we concentrate on the analysis of the data from isolated whole frame losses.

We create six different realizations of whole frame loss events of the 20-minute video, producing 900 distinct isolated whole frame losses. 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, including an introductory session and a break. When viewers see a glitch, they respond to that glitch by pressing the space bar. If the response time is within 2 seconds of the loss, the loss 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 frame loss is calculated as the number of people who see the loss artifact divided by 10. 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. 1800 ground truth visibility scores are obtained (900 for the JM decoder and 900 for the FFMPEG decoder).

VII.B Data analysis

In this section, we compare the visual performance of frame copy (JM) and frame interpolation (FFMPEG).

Figures VII.1(a) and VII.1(b) show the histograms of the visibility of the JM decoder and the FFMPEG decoder, respectively. For the JM decoder, 40.78% of the losses are not observed by any subjects (visibility is zero). For the FFMPEG decoder, 38.89% of the losses are not observed by any subjects. In other words, more than 1/3 of losses are not seen by any user. And for the JM decoder, 62.43% of losses have visibility less than or equal to 0.2, whereas for the FFMPEG decoder, 58.29% of losses have visibility less than or equal to 0.2. For both decoders well over half of isolated whole B frame losses are seen by 2 or fewer out of 10 people. 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 viewers were not perceiving losses in each time slot.

Figure VII.2 is the 3-D histogram of the 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. Many losses are of zero visibility for both FFMPEG and JM decoders, and it is rare that one loss is highly visible in one decoder and of lower visibility in the other. Most of the time, the visibility of a particular whole frame loss is similar (not exactly the same) for different concealment methods. The correlation of the visibility scores between JM and FFMPEG is 0.6043. This motivates us to develop one model to predict the whole frame packet loss visibility for both JM and FFMPEG decoders. We discuss it in the next section.

Also, we want to know whether one decoder is better than the other in terms of whole frame error concealment visually. We start with a simple paired comparison of the ground truth loss visibility scores between JM and FFMPEG. We say a decoder wins if the ground truth of one decoder is lower (visually better) than the other, and loses if it is higher. The result shows that the fractions of JM wins, FFMPEG wins and ties are 33.16%, 29.64% and 37.18%. This means more than 1/3 of the whole frame losses are observed by exactly the same number of observers for both error concealment methods used. Among the tie cases, 79.05% represent losses with zero visibility for both JM and FFMPEG. Also JM wins more times against FFMPEG. When JM conceals the whole frame loss by frame copy, there are no spatial concealment artifacts; it is just a copy of the previous intact frame. However, for FFMPEG that conceals by temporal interpolation, ghosting artifacts may appear when there is enough motion. A visual example is demonstrated in Figure VII.3. Frame 35 is lost and concealed by JM with frame copy in Figure VII.3(a) and by FFMPEG with temporal frame interpolation in Figure VII.3(b). The average whole frame loss visibility over all the data is 0.1716 for JM and 0.1879 for FFMPEG, indicating that on average, the whole frame losses concealed by JM are 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 Figures VII.1(a) and VII.1(b), their distribution is far from normal. Therefore we resort to nonparametric hypothesis testing. The Wilcoxon Signed Rank Test (paired comparison) [90] compares paired data x and y in a two-sided test where the null hypothesis H_0 is that the median of $x - y$ comes from a continuous, symmetric distribution with zero median, 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,

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

distributes approximately as Normal(0,1) when $n > 12$. The p-value is 0.176 ($> 5\%$), meaning that we can not reject the null hypothesis at 95% confidence level that the visibility scores of FFMPEG minus JM come from a distribution of zero median.

VII.C Whole frame packet loss visibility model

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

VII.C.1 Factors extractable from the bitstream

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 were shown to be important to the prediction of packet loss visibility in previous chapters. We consider the residual energy distribution of the MBs in a frame, denoted by **RSENGY**. We take the average of the residual energy of all the MBs in a frame. We denote this quantity as **Mean**RSENGY. **Max**RSENGY denotes the maximal residual energy after motion compensation among all MBs in a frame. **Var**RSENGY denotes the variance of the residual energy of MBs in a frame. Aside from these which were used in previous chapters, here we include two more descriptions of the distribution. The skewness [90] of RSENGY describes the amount of asymmetry of the RSENGY distribution, denoted as **Skew**RSENGY, and the entropy [91] of RSENGY captures the randomness of the RSENGY distribution, denoted as **Ent**RSENGY.

In addition to RSENGY, the **QP** distribution used for each MB is also included. In H.264, the partition of a MB is supported, so the **Interparts** distribution of MBs in a frame is included as a factor. Another important factor involves motion vectors. **MotX** and **MotY** are motion vectors distributions in x and y directions of MBs in each frame. **MotM**, the motion magnitude distribution of MBs in a frame, is considered. To compute the factors related to phase of motion vectors, we only consider macroblocks with non-zero motion, for which the phase is well defined. We denote the phase information distribution of the motion vectors as **MotA**. The packet size distribution in bits in a frame, denoted as **SliceSize**, is also included for prediction.

For each one of these distributions (QP, Interparts, MotX, MotY, MotM, MotA and SliceSize), we include the Mean, Max, Var, Skew and Ent (as we do for RSENGY) as predictive features in our model. In addition, we are interested in how the way MBs are coded can affect the frame loss visibility, thus we include the number of MBs in a frame that are coded in the mode of INTRA (**NumIntraMB**), INTER (**NumInterMB**), DIRECT (**NumDirectMB**) and SKIP (**NumSkipMB**) into factor consideration.

For residual energy, as in Chapter II, 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. Also note for the motion information to be self-contained in a packet, as in Chapter V, the MB coded in “direct” mode should estimate the true motion vectors from neighboring blocks.

VII.C.2 Modeling process and discussion

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 follow the same model developing process as in Chapter V which uses the concept of the M-estimator to account for the effects of outliers. We use the factor set described in Section VII.C.1, plus interaction terms between any two factors in the set by multiplication between two factors.

From Section VII.B we know that the concealed result for JM is not significantly better than for FFMPEG, and that a whole frame loss with high visibility for one decoder is very likely to be highly visible for the other decoder, therefore it is reasonable to make one generalized model for both decoders. One can make such a model by taking the average of the two visibility scores associated with the same whole frame loss. We denote the result **Avg_JM_FFMPEG**. Another way is taking the maximum of the two visibility scores of the JM and FFMPEG decoders; this aims to predict the visibility for the worst decoder for a loss, and we denote the result **Max_JM_FFMPEG**.

Figures VII.4(a) and (b) show decreasing M-estimator as we add factors in the order of importance into the models that predict Avg_JM_FFMPEG and Max_JM_FFMPEG respectively. The circle markers in the plots consider all the factors discussed in Section VII.C.1. We observe that adding more factors in the model produces diminishing returns. In fact, most of those factors involve the computation of skewness and entropy, which are very complicated. Therefore, we remove the factors involving skewness and entropy from consideration. The factors in Figure VII.4(a) and (b) that are marked by diamonds do not include skewness and entropy. We can see that by saving the computation and reducing

Table VII.1: Table of factors for Avg_JM_FFMPEG model in the order of importance.

Order	Factors	Coefficients
α	1	-2.3502
1	MeanMotM	8.5907e-2
2	VarMotY	-2.4423e-3
3	log(MaxRSENGY $+10^{-7}$)	5.7905e-2
4	VarMotX	-7.5725e-4
5	MeanSliceSize × VarMotY	4.8017e-7
6	NumInterMB	-6.0581e-4
7	MaxMotM	3.6750e-3

the number of factors in the model, we only lose 12.4% for (a) and 6.45% for (b) of the full performance achieved by all the circled factors (computed by the M-estimator decrease from the best diamond model to the best circle model, divided by the M-estimator decrease from the initial model to the best circle model). 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 Table VII.1 and Table VII.2, respectively. One interesting observation is that the first four important factors are the same for both models. Also, the information relating to motion vectors is very important; 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 of concealment by both the JM and FFMPEG decoders.

VII.D Conclusion

We present a subjective test and its results on whole B frame loss visibility of the H.264 encoded bitstream. We compare the visual result of the concealment by the JM standard and FFMPEG decoders. For whole frame loss, JM produces frame copy artifact, while FFMPEG produces temporal frame interpolation arti-

Table VII.2: Table of factors for Max_JM_FFMPEG model in the order of importance.

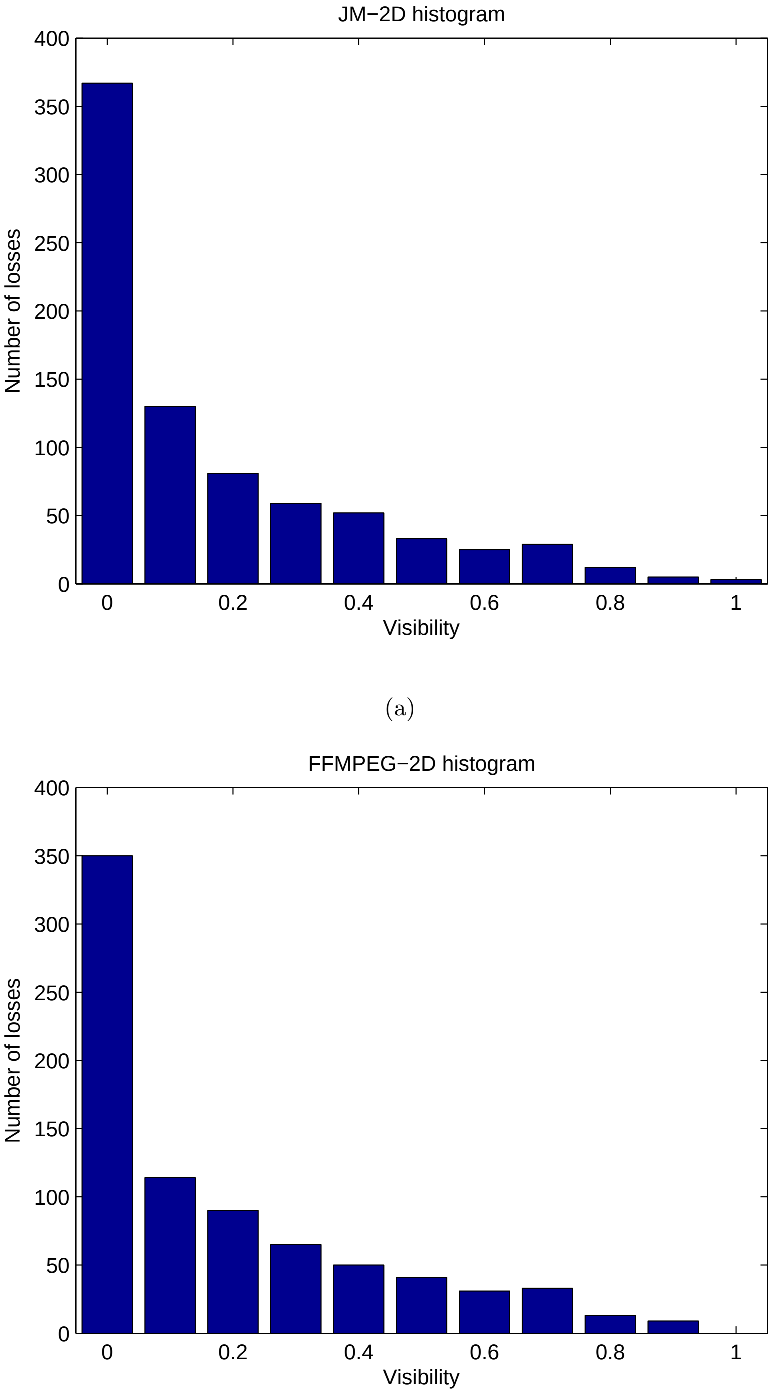
Order	Factors	Coefficients
α	1	-1.930
1	MeanMotM	9.4313e-2
2	VarMotY	-2.2636e-3
3	log(MaxRSENGY $+10^{-7}$)	5.5021e-2
4	VarMotX	-8.3054e-4
5	MaxMotM	9.2753e-3
6	MaxMotY	-6.0405e-3
7	MeanSliceSize × VarMotY	3.9402e-7
8	NumInterMB	-5.1083e-4
9	MaxMotX	-4.4854e-3

fact. We found that there is no statistically significant difference in the visibility of these losses between the two different decoders. Experimental results showed that approximately 40% of all isolated losses were not observed by any viewers, and about an additional 20% of the loss events were only observed by 1 or 2 out of 10 observers. We then developed two whole frame loss visibility models; one predicts the average visibility by the decoders, the other is for the worst case visibility.

VII.E Acknowledgements

This work was supported by Futurewei Technologies, Inc. and by the Center for Wireless Communications at UCSD.

Chapter VII 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 Conference 2010.* Co-author Prof. Cosman directed and supervised the research which forms the basis for Chapter VII. Co-author Y.-L. Chang also contributed to the ideas in this work.



(a)

(b)

Figure VII.1: (a) Histogram of whole frame loss visibility by JM decoder, (b) Histogram of whole frame loss visibility by FFMPEG decoder.

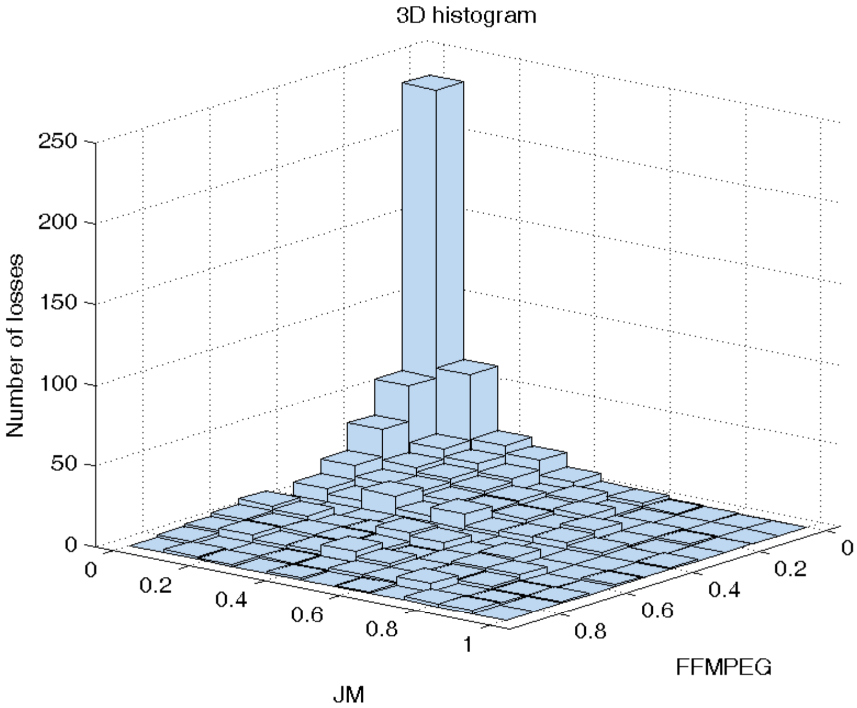
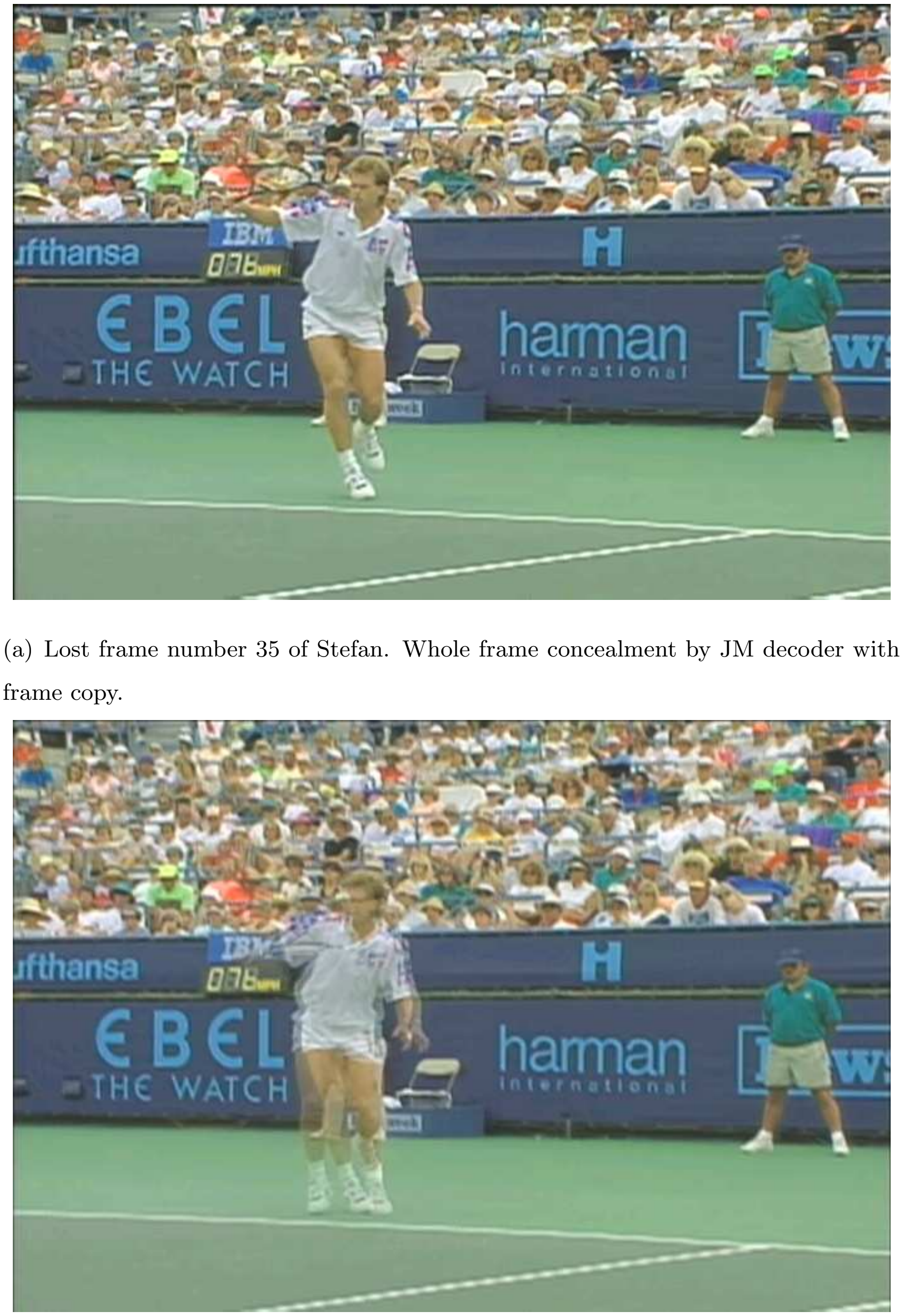


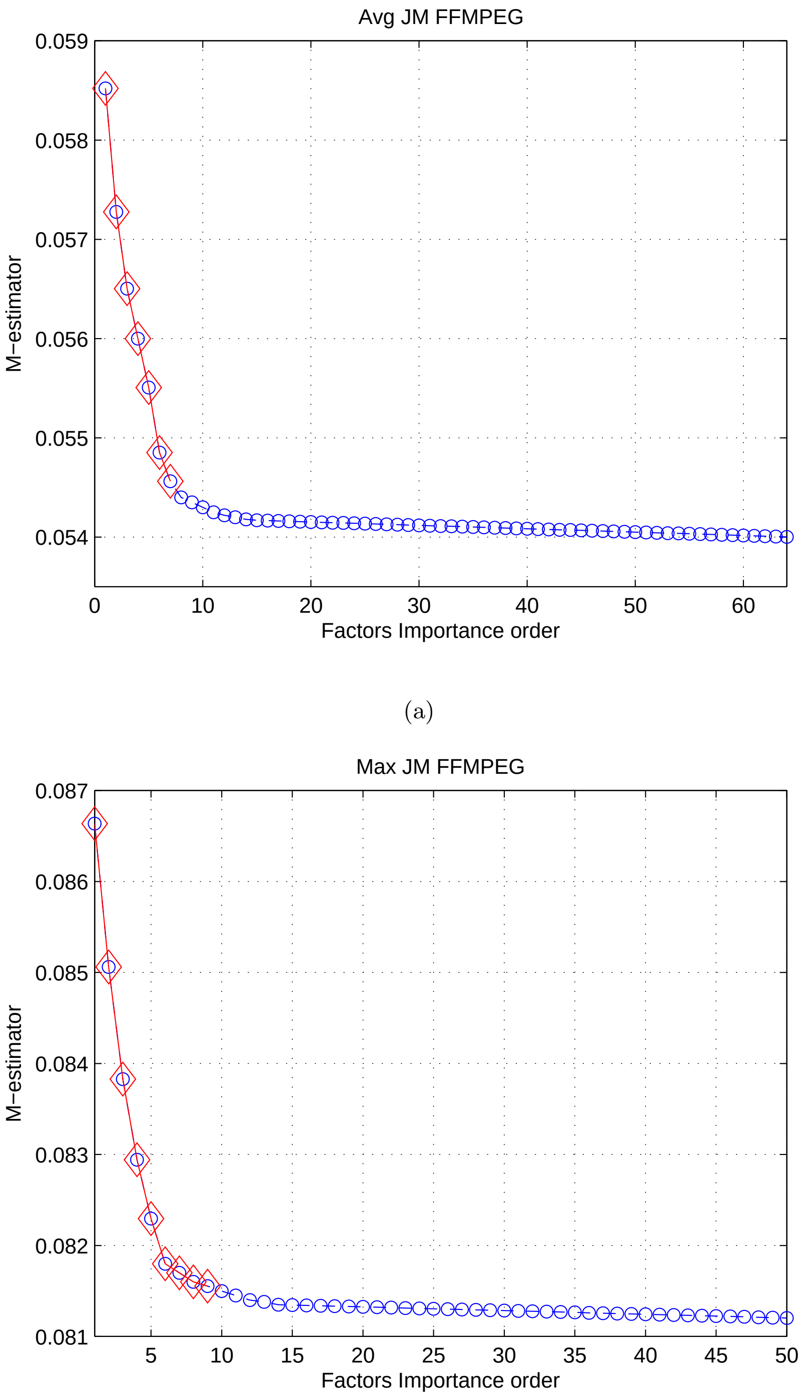
Figure VII.2: 3-D Histogram of whole frame loss visibility by JM decoder and FFMPEG decoder.



(a) Lost frame number 35 of Stefan. Whole frame concealment by JM decoder with frame copy.

(b) Lost frame number 35 of Stefan. Whole frame concealment by FFMPEG decoder with temporal frame interpolation.

Figure VII.3: Frame 35 is lost and concealed by JM decoder with frame copy in (a) and by FFMPEG decoder with temporal frame interpolation in (b).



(a)

(b)

Figure VII.4: The M-estimator plot versus the number of included factors predicting (a) Avg_JM_FFMPEG (b) Max_JM_FFMPEG.

VIII

Conclusion

In this dissertation, we have proposed an encoder-based packet loss visibility model, a network-based packet loss visibility model, and a network-based whole frame loss visibility model. We discuss their applications to packet prioritization, unequal error protection and intelligent packet dropping. We compare our methods to existing ones.

In Chapter II, we discuss an encoder-based packet loss visibility model. The model has broad applicability since it is developed on datasets from multiple subjective experiments using different codecs, different encoder settings, and different decoder error concealment strategies. Factors related to scene cuts and camera motion are found to be effective in predicting the visibility of packet loss.

In Chapter III, we discuss the application of the encoder-based packet loss visibility model to packet prioritization. We use the visibility model to prioritize video packets and use this for perceptual-quality based packet discarding. The proposed policy performs better than the policy using cumulative MSE prioritization in most cases, and outperforms the widely-implemented Drop-Tail in all cases under diverse network conditions and GOP structures. The experiments show that the model performs well for videos with various encoding rates, even though the model is designed for high-quality video transported over a mostly reliable network.

In Chapter IV, we discuss the application of the encoder-based packet loss visibility model to unequal error protection. We use the branch and bound method, K-means clustering and the subgradient method to solve the RCPC channel rate allocation problem to reduce the end-to-end packet loss visibility over an AWGN channel. The subgradient method is most efficient for the optimal channel code rate allocation. We exploit the options of not coding and not sending the packets. Our algorithm significantly outperforms an existing approach under different channel conditions, video clips and GOP structures.

In Chapter V, we develop a network-based packet loss visibility model. We propose self-contained and network-based packet loss visibility models for SDTV and HDTV resolutions. These models perform only slightly less well than the much more complicated non-self-contained models that could be implemented only at the encoder. Due to a wider viewing angle for HDTV, the spatial location of the packet loss in HDTV matters more than in SDTV. For both SDTV and HDTV models, the temporal duration of the error propagation is a very important factor for a packet to be visible.

In Chapter VI, we discuss the application of the network-based packet loss visibility model to packet dropping. We use a network-based packet loss visibility model to measure the visual importance of packets incoming to a router. The estimated visibility scores are used by the router to perform intelligent packet dropping. We show that dropping whole frames and concealing by simple frame interpolation produces better quality video than dropping on a packet (slice) basis. The visual advantage of dropping whole frames is sufficiently large that, except for low dropping rates, it pays to drop whole frames even when that means slightly exceeding the target bit reduction rate for the GOP. A simple visibility model that can be implemented inside the network provides a better basis for choosing frames to drop than just targeting B frames for dropping.

In Chapter VII, we develop a network-based packet loss visibility model for whole frame loss. We present a subjective test and its results on whole B frame loss visibility for an H.264 encoded bitstream. There is no statistically significant difference in the visibility of these losses between the JM and FFMPEG decoders. Experimental results show that approximately 40% of all isolated losses were not observed by any viewers, and about an additional 20% of the loss events were only observed by 1 or 2 out of 10 observers. We develop two whole frame loss visibility models; one predicts the average visibility by the decoders, the other is for the worst case visibility.

VIII.A Future work

The future work for the application of the network model includes:

- *Network monitoring*: The network model can serve as a quality monitor for videos transmitted in the network. This can allow one to understand the quality of the video for a specific channel. This is important for, for example, an IPTV provider.
- *Intelligent early dropping*: In this dissertation, we look at methods of packet dropping *during* network congestion. It may also be desirable for the router to perceive the forthcoming congestion and drop the visually unimportant packets earlier so that the more important packets can be saved and scheduled for transmission even during congestion.
- *Fair dropping among streams*: When there are multiple video flows into the router, we should define a best way to drop packets among different streams. One way is to drop the packets to maximize the sum of video quality across all users. Another approach is to drop packets so that largest video quality degradation among the streams is minimized. Factors such as pricing can also come into play.

For the subjective experiment of whole frame losses, several topics can be extended:

- *Whole frame dropping protocol*: In the whole frame dropping method proposed in the dissertation, the importance of the whole frame drop is estimated by the summation of the visibility scores of all the packets in the frame. By the network-based whole frame model, we can estimate directly the visual importance of the whole frame loss. And based on this measurement, we can develop a dropping method specifically for whole frame loss.

- *Encoder-based whole frame loss visibility model*: By including effective factors in the pixel domain, such as MSE and scene cut information, the prediction accuracy can increase. Based on this model, we can perform frame level *unequal error protection* and *prioritization.*

Bibliography

[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13, Apr 2004.

[2] Z. Wang, L. Lu, and A. C. Bovik. Video quality assessment based on structural distortion measurement. *Signal Processing: Image Communication, Special issue on objective video quality metrics*, 19(2), Feb 2004.

[3] M. A. Masry and S. S. Hemami. A metric for continuous quality evaluation of compressed video with severe distortions. *Signal processing: Image communication*, 19(2), Feb 2004.

[4] VQM software. [Online]. Available:
http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm.

[5] 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.

[6] S. Wolf and M. H. Pinson. Low bandwidth reduced reference video quality monitoring system. *First International Workshop on Video Processing and Quality Metrics*, Jan 2005.

[7] I. P. Gunawan and M. Ghanbari. Reduced-reference video quality assessment using discriminative local harmonic strength with motion consideration. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(1):71–83, January 2008.

[8] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi. Perceptual blur and ringing metrics: Application to JPEG2000. *Signal processing: Image communication*, pages 163–172, Feb 2004.

[9] A. Eden. No-reference estimation of the coding PSNR for H.264-coded sequences. *IEEE Transactions on Consumer Electronics*, 53:667 – 674, May 2007.

[10] A. M. Eskicioglu and P. S. Fisher. Image quality measures and their performance. *IEEE Transactions on Communications*, 43(12):2959–2965, 1995.

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

[12] T. Liu, Y. Wang, J. M. Boyce, Z. Wu, and H. Yang. Subjective quality evaluation of decoded video in the presence of packet losses. *ICASSP. IEEE*, pages 1125–1128, April 2007.

[13] Y. J. Liang, J. G. Apostolopoulos, and B. Girod. Analysis of packet loss for compressed video: does burst-length matter? *ICASSP. IEEE*, 5:684–687, 2003.

[14] S. Tao, J. Apostolopoulos, and R. A. Guerin. Real-time monitoring of video quality in IP networks. *Proceedings NOSSDAV'05*, pages 129–134, 2005 June.

[15] A. R. Reibman, V. Vaishampayan, and Y. Sermadevi. Quality monitoring of video over a packet network. *IEEE Transactions on Multimedia*, 6(2):327–334, Apr 2004.

[16] G. W. Cermak. Videoconferencing service quality as a function of bandwidth, latency, and packet loss. *T1A1.3/2003-026, Verizon Laboratories*, May 2003.

[17] B. Chen and J. Francis. Multimedia performance evaluation. *AT&T Technical Memorandum*, February 2003.

[18] S. Mohamed and G. Rubino. A study of real-time packet video quality using random neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(12):1071–1083, Dec 2002.

[19] 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*, 2(2):212–222, April 1993.

[20] R. V. Babu, A. S. Bopardikar, A. Perkis, and O. I. Hillestad. No-reference metrics for video streaming applications. *International Workshop on Packet Video*, Dec 2004.

[21] H. Rui, C. Li, and S. Qiu. Evaluation of packet loss impairment on streaming video. *Journal of Zhejiang University SCIENCE*, 7, Apr 2006.

[22] S. Winkler and R. Campos. Video quality evaluation for internet streaming applications. *SPIE, Human Vision and Electronic Imaging VIII*, 5007:104–115, Jan 2003.

[23] R. R. Pastrana-Vidal and J.-C. Gicquel. Automatic quality assessment of video fluidity impairments using a no-reference metric. *International Workshop on Video Processing and Quality Metrics*, Jan 2006.

[24] A. R. Reibman, S. Kanumuri, V. Vaishampayan, and P. C. Cosman. Visibility of individual packet losses in MPEG-2 video. *IEEE ICIP*, October 2004.

[25] S. Kanumuri, P. C. Cosman, and A. R. Reibman. A generalized linear model for MPEG-2 packet-loss visibility. *International Packet Video Workshop*, December 2004.

[26] S. Kanumuri, P. C. Cosman, A. R. Reibman, and V. Vaishampayan. Modeling packet-loss visibility in MPEG-2 video. *IEEE Transactions on Multimedia*, 8:341–355, Apr 2006.

[27] S. Kanumuri, S. G. Subramanian, P. C. Cosman, and A. R. Reibman. Packet-loss visibility in H.264 videos using a reduced reference method. *IEEE ICIP*, Oct 2006.

[28] A. R. Reibman and D. Poole. Characterizing packet loss impairments in compressed video. *IEEE ICIP*, Sept 2007.

[29] A. R. Reibman and D. Poole. Predicting packet-loss visibility using scene characteristics. *International Packet Video Workshop*, pages 308–317, Sept 2007.

[30] J.C. De Martin and D. Quaglia. Distortion-based packet marking for MPEG video transmission over diffserv networks. *ICME*, pages 111 – 116, Oct 2001.

[31] F. De Vito, L. Farinetti, and J.C. De Martin. Perceptual classification of MPEG video for differentiated-services communications. *ICME*, 1:141–144, Aug 2002.

[32] D. Quaglia and J. C. De Martin. Adaptive packet classification for constant perceptual quality of service delivery of video streams over time-varying networks. *ICME*, 3:369–72, July 2003.

[33] J. Chakareski, J. Apostolopoulos, and B. Girod. Low-complexity rate-distortion optimized video streaming. *International Conference on Image Processing*, 2004.

[34] J. Chakareski, J.G. Apostolopoulos, S. Wee, and B. Girod. Rate-distortion hint tracks for adaptive video streaming. *IEEE Transactions on Circuits and Systems for Video Technology*, 15:1257–1269, Oct. 2005.

[35] J. Chakareski and P. Frossard. Rate-distortion optimized bandwidth adaptation for distributed media delivery. *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2005.

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

[37] W. Tu, W. Kellerer, and E. Steinbach. Rate-distortion optimized video frame dropping on active network nodes. *Packet Video*, 2004.

[38] W. Tu, J. Chakareski , and E. Steinbach. Rate-distortion optimized frame dropping and scheduling for multi-user conversational and streaming video. *Journal of Zhejiang University - Science A*, 2006.

[39] NS Project. [Online]. Available: http://www.isi.edu/nsnam/ns/.

[40] M. Bystrom and J. W. Modestino. Combined source-channel coding schemes for video transmission over an additive white Gaussian noise channel. *IEEE JSAC*, 18(6):880–890, Jun 2000.

[41] F. Zhai, Y. Eisenberg, T. N. Pappas, R. Berry, and A. K. Katsaggelos. Rate-distortion optimized hybrid error control for real-time packetized video transmission. *IEEE Trans. Image Proc.*, pages 40–53, Jan 2006.

[42] C.-L. Huang and S. Liang. Unequal error protection for MPEG-2 video transmission over wireless channels. *Signal Processing: Image Communication*, 19:67–79, Jan. 2004.

[43] O. Harmanci and A. M. Tekalp. Rate-distortion optimal video transport over IP allowing packets with bit errors. *IEEE Trans. on Image Proc.*, 16(5):1315–1326, May 2007.

[44] N. Thomos, S. Argyropoulos, N. V. Boulgouris, and M. G. Strintzis. Robust transmission of H.264/AVC video using adaptive slice grouping and unequal error protection. *IEEE International Conference on Multimedia and Expo*, pages 593–596, July 2006.

[45] J.Y. Shih and W.J. Tsai. A new unequal error protection scheme based on FMO. *ICIP*, pages 3068–3071, 2008.

[46] F. Marx and J. Farah. A novel approach to achieve unequal error protection for video transmission over 3G wireless networks. *Signal Processing: Image Communication*, 19(4):313 – 323, 2004.

[47] C. Dubuc, D. Boudreau, and F. Patenaude. The design and simulated performance of a mobile video telephony application for satellite third-generation wireless systems. *IEEE Trans. Multimedia*, pages 424–431, Dec 2001.

[48] T. Fang and L.-P. Chau. A novel unequal error protection approach for error resilient video transmission. *IEEE International Symposium on Circuits and Systems*, 4:4022– 4025, May 2005.

[49] K. Stuhlmuller, N. Farber, M. Link, and B. Girod. Analysis of video transmission over lossy channels. *IEEE JSAC*, pages 1012–1032, June 2000.

[50] Q. Qu, Y. Pei, and J. W. Modestino. An adaptive motion-based unequal error protection approach for real-time video transport over wireless IP networks. *IEEE Trans. on Multimedia*, pages 1033 – 1044, October 2006.

[51] Q. Qu, Y. Pei, and J. W. Modestino. Robust H.264 video coding and transmission over bursty packet-loss wireless networks. *IEEE Vehicular Technology Conference*, 5:3395–3399, October 2003.

[52] G. Baruffa, P. Micanti, and F. Frescura. Error protection and interleaving for wireless transmission of JPEG 2000 images and video. *IEEE Trans. Image Proc.*, 18(2):346–356, Feb 2009.

[53] A. E. Mohr, E. A. Riskin, and R. E. Ladner. Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction. *IEEE JSAC*, pages 819–828, Jun 2000.

[54] T.-L. Lin and P. Cosman. Optimal RCPC channel rate allocation in AWGN channel for perceptual video quality using integer programming. *First International Workshop on Quality of Multimedia Experience (Qomex), IEEE*, 2009.

[55] B. Krishnamoorthy. Bounds on the size of branch-and-bound proofs for integer knapsacks. *Operations Research Letters*, 36(1):19 – 25, 2008.

[56] T.-L. Lin and P. Cosman. Perceptual video quality optimization in AWGN channel using low complexity channel code rate allocation. *Asilomar Conference on Signals, Systems and Computers*, 2009.

[57] Y.-Z. Huang and J. G. Apostolopoulos. A joint packet selection/omission and FEC system for streaming video. *IEEE ICASSP*, 1:I–845–I–848, April 2007.

[58] M. Ardito, M. Gunetti, and M. Visca. Influence of display parameters on perceived HDTV quality. *IEEE Transactions on Consumer Electronics*, 42:145–155, Feb 1996.

[59] S. Pechard, M. Carnec, P. Le Callet, and D. Barba. From SD to HD television: effects of H.264 distortions versus display size on quality of experience. *ICIP*, 2006.

[60] F. Boulos, W. Chen, B. Parrein, and P. Le Callet. A new H.264/AVC error resilience model based on regions of interest. *Packet Video*, June 2009.

[61] M. Naccari, M. Tagliasacchi, and S. Tubaro. No-reference video quality monitoring for H.264/AVC coded video. *IEEE Transactions on Multimedia*, 11(5):932 – 946, Aug. 2009.

[62] S. Tao, J. Apostolopoulos, and R. Guerin. Real-time monitoring of video quality in IP networks. *IEEE/ACM Transactions on Networking*, 16(5):1052 – 1065, Oct. 2008.

[63] 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.

[64] H.264/AVC JM software : http://iphome.hhi.de/suehring/tml/.

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

[66] 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*, 5292:182–193, 2004.

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

[68] P. McCullagh and J. A. Nelder. Generalized linear models 2^{nd} edition. *Chapman & Hall*, 1989.

[69] D. W. Hosmer and S. Lemeshow. Applied logistic regression, 2nd ed. *Wiley-Interscience*, 2000.

[70] W. Rey. *Introduction to robust and quasi-robust statistical methods.* Springer, 1983.

[71] L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and regression trees. *Wadsworth, Pacific Grove, CA*, 1984.

[72] J. H. Friedman and N. I. Fisher. Bump hunting in high-dimensional data. *Statistics and Computing*, 9:123–143, 1999.

[73] Y. Sermadevi and A. R. Reibman. Unpublished subjective test results. Sept 2002.

[74] O. Nemethova, M. Ries, M. Zavodsky, and M. Rupp. PSNR-based estimation of subjective time-variant video quality for mobiles. *Proc. of the MESAQUIN*, 2006.

[75] A. Hanjalic. Content-based analysis of digital video. *Kluwer Academic Publishers, Boston*, 2004.

[76] Z. Liu, D. Gibbon, E. Zavesky, B. Shahraray, and P. Haffner. AT&T Research at TRECVID. 2006.

[77] Y.-P. Tan, D. D. Saur, S. R. Kulkami, and P. J. Ramadge. Rapid estimation of camera motion from compressed video with application to video annotation. *IEEE Transactions on Circuits and Systems for Video Technology*, 10(1):133–146, Feb 2000.

[78] K. Stuhlmuller, N. Farber, M. Link, and B. Girod. Analysis of video transmission over lossy channels. *IEEE Journal on Selected Areas in Communication*, 18(6):1012–1032, June 2000.

[79] M. Budagavi and J. D. Gibson. Multiframe video coding for improved performance over wireless channels. *IEEE Transactions on Image Processing*, 10(2):252–265, February 2001.

[80] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: data mining, inference and prediction. *Springer-Verlag, New York*, 2001.

[81] J. F. Kurose and K. W. Ross. *Computer networking: a top-down approach featuring the internet.* Addison Wesley, 3rd edition, 2004.

[82] D. P. Bertsekas. *Nonlinear programming.* Athena Scientific, 2nd edition.

[83] D. Li and X. Sun. *Nonlinear integer programming.* International Series in Operations Research and Management Science. Springer, 2006.

[84] S. Boyd and L. Vandenberghe. *Convex optimization.* Cambridge University Press, 2004.

[85] S. L. P. Yasakethu, W. A. C. Fernando, S. Adedoyin, and A. Kondoz. A rate control technique for off line H.264/AVC video coding using subjective quality of video. *Consumer Electronics, IEEE Transactions on*, 54:1465–1472, Aug. 2008.

[86] ITU-R BT.710-4 subjective assessment methods for image quality in high-definition television. Jan 1998.

[87] DSL forum technical report TR-126: Triple-play services quality of experience (QoE) requirements. Dec 2006.

[88] G. Mullet. Why regression coefficients have the wrong sign. *Journal of Quality Technology*, 8(3), 1976.

[89] R. Sharpe, D. Zriny, and D. De Vleeschauwer. Alcatel-Lucent technical paper : access network enhancements for the delivery of video services. May 2005.

[90] R. Larsen and M. Marx. *An introduction to mathematical statistics and its applications.* Pearson Edu, 4th edition.

[91] R. C. Gonzalez and R. E. Woods. *Digital image processing.* Prentice Hall, 2nd edition.