Title

A Versatile Model for Packet Loss Visibility and its Application to Packet Prioritization

Journal

IEEE Transactions on Image Processing, 19(3)

ISSN

1057-7149 1941-0042

Authors

Lin, Ting-Lan
Kanumuri, S.
Zhi, Yuan
et al.

Publication Date

2010-03-01

DOI

10.1109/TIP.2009.2038834

Peer reviewed

A Versatile Model for Packet Loss Visibility and its Application to Packet Prioritization

Ting-Lan Lin, *Student Member, IEEE*, Sandeep Kanumuri, *Member, IEEE*, Yuan Zhi, David Poole, Pamela C. Cosman, *Fellow, IEEE*, and Amy R. Reibman

***Abstract*—In this paper, we propose a generalized linear model for video packet loss visibility that is applicable to different group-of-picture structures. We develop the 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. We consider factors not only within a packet, but also in its vicinity, to account for possible temporal and spatial masking effects. We discover that the factors of scene cuts, camera motion, and reference distance are highly significant to the packet loss visibility. We apply our visibility model to packet prioritization for a video stream; when the 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. To show the effectiveness of our visibility model and its corresponding packet prioritization method, experiments are done to compare our perceptual-quality-based packet prioritization approach with existing Drop-Tail and Hint-Track-inspired 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. Our model was developed using data from high encoding-rate videos, and designed for high-quality video transported over a mostly reliable network; however, the experiments show the model is applicable to different encoding rates.**

***Index Terms*—Packet dropping policy, packet loss, perceptual video quality, video coding, visibility model.**

I. Introduction

TRANSMISSION of compressed video over a network is becoming more and more popular due to the rising demand for multimedia applications. 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 in the network to help evaluate the quality of video reception. A good in-the-network 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 in the network 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 or the decoded pixels without reference video information. One of the most widely used FR metrics is MSE (Mean square 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, is shown to be better correlated with human perception than other FR video quality metrics [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 correlations 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 (for example, due to congestion) can significantly damage video quality during transmission. 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, they are commonly used to measure video quality. The relation between PSNR and perceptual quality scores is considered in [10]. 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 [11]. Average

Manuscript received December 29, 2008; revised August 17, 2009. First published December 18, 2009; current version published February 18, 2010. This work was supported in part by the National Science Foundation under Grant 0635165. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Minh N. Do.

T.-L. Lin and P. C. Cosman are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: tinglan@ucsd.edu; pcosman@ucsd.edu).

S. Kanumuri is with the DoCoMo Communications Laboratories USA, Inc., Palo Alto, CA 94304 USA (e-mail: skanumuri@docomolabs-usa.com).

Y. Zhi is with the Texas Instruments, Inc., Stafford, TX 77477-3099 USA (e-mail: yuanz@ti.com).

D. Poole and A. R. Reibman are with the AT&T Labs-Research, Florham Park, NJ 07932 USA (e-mail: amy@research.att.com; poole@research.att.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2009.2038834

performance across an entire video sequence is the focus in [12], 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 [13] are developed to estimate the MSE caused by a packet loss.

Much of the effort to understand the visual impact of packet losses [14]–[17] has focused on modeling the average quality of videos as a function of average packet loss rate (PLR). Video conferencing was studied in [14] using the average consumer judgments on the relative importance of bandwidth, latency and packet loss. A random-neural-network model was developed in [16] to assess quality given different bandwidths, frame-rates, packet loss rates, and I-block refresh rates. In [18] and [19], 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.* [17] 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 [20], 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 [21], 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. For *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 packet visibility model can also be useful for *in-the-network quality monitoring* to obtain accurate and real-time information on the transmitted video.

In our previous work [22], we 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 [23] was applied to classify the losses as visible or invisible. This work was extended in [24] and [25] to model the probability of packet loss visibility using a generalized linear model (GLM) [26]. The visibility for H.264 packets is discussed in [27]. We derived visibility models specifically for individual and multiple packet losses based on RR factors.

In all these studies, the main factors we considered are based on encoding information within a slice (packet), such as motion vectors, residuals, and number of inter partitions. In [28], we focus 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, we considered both the temporal and spatial edges induced by a packet loss, and also the error duration. In addition, in [28], we 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. In [29], we 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) [30] was used to understand when the packet loss will be very visible and very invisible.

In this paper, we present a new packet-loss visibility model based on a more general strategy for factor inclusion than in [28], [29]. The goals of this paper are twofold: to develop a visibility model for different GOP structures and encoding rates, and to demonstrate the effectiveness of a packet prioritization application based on our visibility model. 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 [31], [32], network status and end-to-end QOS constraint [33]. Also, the Rate-Distortion Hint-Track method was proposed for packet scheduling [34], [35] and packet dropping [36], [37]. Especially in [36] and [37], 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 [38] using a rate vector and a distortion matrix, and employing a different optimization philosophy. A detailed discussion of Hint-Track (HT) and Distortion Matrix (DM) methods is in [39]. 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; our 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 NS-2 (Network Simulator) [40]. The comparisons are made using the well-known perceptual quality metric VQM [4]. Some of the results in this paper were presented in [41].

There are many differences between this paper and our previous work which appeared in [25]. Our earlier work used data from a single codec (MPEG-2), video resolution (720×480), GOP structure (IBBP), and error concealment method (ZMEC). The current paper uses data from multiple codecs, video resolutions, GOP structures, and error concealment methods. Also, the visibility model in [25] uses features which are specific to that GOP structure. For example, the features named P1, P2, P3, and P4 denote P frames with different distances to the I frame for the IBBP GOP structure. These variables may be undefined if there are more or fewer than 4 P frames in a GOP, or if the GOP structure is other than IBBP. In this paper, we avoid GOP-specific variables. 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. Another major difference with [25] is that the current paper demonstrates the utility of this general model in a packet prioritization application; the application uses various GOP structures, including ones which were not used in the subjective experiments used for building the model.

This paper is organized as follows: Section II describes the experiment settings for the three different subjective tests and the variety of settings used for video encoding. Section III discusses possible applications of the proposed model. Section IV 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 V, we first provide a brief introduction to the GLM modeling method. Then we illustrate our GLM model building strategy using all the different data sets and incorporate significant factors. Section VI presents the experiment results comparing our visibility-based prioritization method with others. Section VII concludes the paper.

II. Subjective Datasets

The major purpose of this work is to develop a generalized and robust visibility model for packet loss impairments. Therefore, we combine the results of three prior subjective experiments [25], [27], [42] in which the video clips are generated by using various codecs and settings as summarized in Table I. The data sets we used are the same as in [29], and the description of the data sets in this section is mostly based on [29].

Tests 1 and 2 use videos compressed by MPEG-2 at spatial resolution 720×480 with an adaptive GOP (group-of-picture) structure in which an I-frame is inserted at each scene cut. In these videos, there are usually 2 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, we explore generic packet sizes, 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).

TABLE I
Summary of Subjective Tests' Parameters and Their Datasets

	Test 1 [42]		Test 2 [25]	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

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 that is designed for speed rather than error resilience. Such a decoder will effectively "conceal" missing data in a reference frame using data from *two* reference frames ago, while missing data from B-frames are "concealed" using data from the prior B-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 [22], [24], [25]. 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 I. 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 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 our subjective experiments is to obtain the ground truth on the visibility of packet losses. In each of our 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 s of every 4-s 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 allow us to understand the visibility of individual packet losses. However, we will show in the experimental section that our model is robust to various packet loss rates and to losses which may not be isolated. The distribution of the losses in three tests are different. In Test 1, we forced roughly 1/7th of all losses to be in B-frames, 1/7th in I-frames, and 5/7th in P-frames, and we also forced roughly 20% of losses to cause an entire frame to be lost. In Test 2, we have 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 is an expert viewer. A 1-min pilot training video is 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 six 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 is processed as in [25] 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.

III. Applications of a Packet-Loss Visibility Model

Before describing what we can measure about a packet loss to predict its visibility, it is worthwhile to describe briefly several applications of our visibility model.

In one scenario, our visibility model is used for in-network quality monitoring of transmitted video, as described in [25]. In this application, the visibility model is computed for the specific loss pattern that is observed in the network. Useful NR factors are extracted from the actual lossy bitstream observed in the network, while any required RR factors are sent to the quality monitor on a reliable side channel. System constraints dictate that there should not be FR factors.

In a second scenario, explored in this paper in Section VI, our visibility model is used to prioritize packets for transmission by a video server. In this case, the goal is to label each packet with a priority, assigned using our visibility model, that describes 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 precomputed 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 paper, we assume that 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.

Whether computing factors for actual losses (for in-network quality monitoring) or for hypothetical losses (for packet prioritization), 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. When computing factors for hypothetical losses, either for the RR factors for in-network monitoring or for the packet priority assignment, we assume that the neighboring packets are not lost. Since this will not be true in the decoder when the surrounding packets are also lost, per-loss factors may not be completely accurate.

In Section IV, when we describe the factors we choose to predict the visibility of packet loss, we indicate for each factor both whether it must be computed at the encoder or could be computed at the server for the application of prioritized transmission, as well as whether it is a RR or NR factor for the application of in-network quality monitoring. To enable the model to be used for both applications, we do not consider FR factors.

IV. 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, we explore these issues by describing attributes that affect the visibility of packet loss impairments, and describe the associated measurements, or factors. To facilitate the following discussion, we define:

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

A. Encoded Signal at Location of Loss

Here we first describe the attributes of the encoded signal *at the location of the packet loss*. 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.

We consider motion information to be an underlying feature of a video, independent of the compression algorithm. Therefore, we measure the following RR signal descriptors related to motion information directly from the *uncompressed* signal $f(t)$. For each macroblock, we measure its motion vector (x,y) by forward motion estimation from the previous frame. For each packet, we define **MOTX** and **MOTY** to be the mean motion vector in the x and y directions over all MBs in the packet. We also compute **MotionVarX** and **MotionVarY**, the variance of the the motion vectors in the x and y directions over the macroblocks in the packet, and define a high-motion descriptor **HighMOT** to be true if $\mathbf{MOTM} = \sqrt{MOTX^2 + MOTY^2} > \sqrt{2}$. **ResidEng** is the average residual energy after motion compensation within a packet. The above motion-related descriptors were also considered in [25]. Finally, **SigMean** and **SigVar** are the mean and variance of the signal $f(t)$.

B. 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 forward masking and it decreases visibility of packet loss. Backward 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 [43]. 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 (EDA). We extract these factors 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 [44] and [45]. We label each packet loss 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. We define **FarConceal** to be true if **MaxDistToRef** $(\text{maximum of } \mid \text{DistToRef} \mid \text{ in a slice}) \geq 3$. In this inequality, MaxDistToRef has units of frames. We also define a Boolean variable, **OtherSceneConceal**, which is TRUE if $\mid \text{DistFromSceneCut} \mid < \mid \text{MaxDistToRef} \mid$, 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, we define a Boolean variable **BeforeSceneCut**, which is TRUE if $-0.4 \text{ s} < \text{DistFromSceneCut} < 0 \text{ s}$ [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, we define the Boolean variable **AfterSceneCut**, which is TRUE when both OtherSceneConceal is FALSE and $0 \text{ s} < (\text{DistFromSceneCut} + \text{Duration}) < 0.25 \text{ s}$.

Camera motion information can also be extracted from the compressed video using a number of techniques, including those

TABLE II
DISTRIBUTION OF CAMERA MOTION IN ORIGINAL CONTENT 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

in [46]. In this paper, we classify scenes based on four camera-motion types: still, panning, zooming, or complex camera motions. Table II 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. We observe that significantly fewer viewers saw packet loss in still scenes than in panning or zooming scenes. Therefore, we define **NotStill** to be TRUE if motion type is not still.

C. 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. We consider **SBM**, Slice Boundary Mismatch, to describe the impact of packet loss on slice boundaries. Methods to measure SBM can be found in [28] and [29].

D. 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 [47]. In addition, using long-term prediction in H.264 can improve error attenuation [48].

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**). From these, we create Boolean variants of these factors: **SXTNT2** is true when two consecutive slices are lost $(\mathrm{SpatialExtent} = 2)$, **SXTNTFrame** when all slices in the frame are lost, and **Error1Frame** is TRUE if the packet loss lasts only one frame $(\mathrm{Duration} = 1)$.

MSE and SSIM (Structural Similarity Index) are commonly used to characterize the amplitude of the error. If we are interested in an accurate evaluation of quality degradation due to both compression artifacts *and* packet loss, then these must be computed at the encoder, since they depend on $f(t)$. However, we choose to consider here only the quality degradation due to packet loss *without* encoding artifacts. Therefore, when calculating MSE and SSIM, we use $\hat{f}(t)$ as 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

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

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 in part, but only indirectly measures attributes such as error size. Also, MSE can neither quantify the spatio-temporal frequency characteristics of the error, nor capture any information about error location or pattern. The SSIM for one frame is defined as

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

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

TABLE III
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

possible packet loss dramatically increases the computational complexity. This is prohibitively expensive, and, thus, we consider neither CumulativeMSE or CumulativeSSIM in our visibility model.

Instead, we calculate only the initial error induced by a packet loss within the frame where the packet loss occurs. 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. We consider here **MaxIMSE**, defined as the maximum per-MB MSE over all MBs in the initial impairment, and **MinISSIM**, 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 variances of the encoded and decoded signals, as well as their MSEs. Table III summarizes the factors.

V. MODELING APPROACHES

Our goal in this paper is to develop a model that predicts the probability of a lost packet being visible to viewers based on the factors discussed above. 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 suitable for our purpose since it can be used to predict the probability parameter of a binomial distribution. In this section, we give a brief description of GLMs and then we present our approach to develop a GLM from three data sets of different sizes. Cross-validation and random seeds are introduced to make the resulting model more robust to potential over-fitting.

In order to reduce the dependency on the RR factors sent from the encoder, as discussed in III, in this work we use only the motion and the residual information. We consider CumulativeMSE and CumulativeSSIM to be too computationally intensive. In [29], two classes of models were developed based on the pooling strategy of initial MSE and initial SSIM (either per-frame averaging or maximum-over-macroblock pooling). In this paper, to obtain the best possible model, we do not make this distinction.

A. Introduction of Generalized Linear Models

GLMs are an extension of classical linear models [26], [49]. 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 [26]. Let $y_1, y_2, \ldots, y_N$ be a realization of independent random variables $Y_1, Y_2, \ldots, Y_N$ 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

$$g(p_i) = \gamma + \sum_{j=1}^{P} x_{ij}\beta_j \tag{3}$$

where $g(.)$ is called the link function, which is typically nonlinear, and $\beta_1, \beta_2, \ldots, \beta_P$ 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

$$g(p) = \log\left(\frac{p}{1-p}\right). \tag{4}$$

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. To obtain the model coefficients for considered factors, an iteratively re-weighted least-squares technique is used to generate a maximum-likelihood estimate. The statistical software R [50] is used for model fitting and analysis. This procedure is also used in [25] and [27].

B. GLM Model Building Approach on Multiple Data Sets

The subjective datasets available for training our 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 we described in Section IV 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 IV. 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 I, we have 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 I by the mean number of viewers who saw each loss). We have most data for the MCEC, which produces fewer noticeable errors. If we train the model using samples chosen randomly from the combined dataset, the resulting fit will be dominated by the MCEC strategy. Therefore, we train models 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 [51] 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 we repeat for different folds with different training and testing partitions of the data. We select our training and testing sets based on the fact that we should achieve equal representation from all datasets including Dataset 1, which has the fewest samples (215). Specifically for each fold, we randomly choose 159 samples from each dataset to fit a model using 159×3 training data. Also, we have a testing set containing the remaining 56 samples from Dataset 1, the remaining 921 samples from Dataset 2, and the remaining 2001 samples from Dataset 3. We apply the method discussed in Section V-A to estimate the model coefficients from the *training set* for given factors, and then evaluate the performance error of the fitted model in the jth fold using the *testing set* as follows:

$$q_j = \frac{1}{3}\sum_{k=1}^{3}\left[\frac{1}{N_k}\sum_{\substack{i\text{th packet loss}\\ \text{in testing set } k}}(p_i - \tilde{p}_i)^2)\right] \tag{5}$$

where $\tilde{p}_i$ is the predicted fraction of viewers who saw the ith packet loss, and N_k is the number of samples in the testing set of Dataset k. We choose four-fold cross-validation: we do the fitting process for a total of four times with four different folds, therefore producing four fitted models and $q_j,\ j = 1,2,3,4$. We repeat this four-fold procedure four times with four different random seeds. We define the average performance error of these sixteen models as Q

$$Q = \frac{1}{16}\sum_{r=1}^{4}\sum_{j=1}^{4} q_j^r \tag{6}$$

where the superscript r stands for the rth random seed.

For factor selection, we use Q to decide if a specific factor is significant and should be included in the model: for each considered factor added to the model, we calculate a Q by the 4-seeds-4-folds GLM modeling process. We *include* a factor only if the model with that factor included has smaller Q than the model without that factor. By the same idea, we *exclude* factors from the model if it has lower Q without them. To obtain the factor coefficients, we use the fitting from the seed that achieved the lowest performance error. The factors and coefficients of our final model are summarized in Table IV. 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.

TABLE IV
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

VI. EXPERIMENTAL RESULTS ON THE APPLICATION OF THE VISIBILITY MODEL TO PACKET PRIORITIZATION

Our 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 our 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.

Several existing packet classification methods were introduced in Section I. Major applications of packet classification are *packet prioritization for a differentiated-services network* [31]–[33], *packet scheduling in the transmitter* [34], [35] 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* [36], [37], 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 [37] 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) into 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 our visibility model that prioritizes packets. At the server, we set a packet to be low priority when its visibility is less than 0.25, and

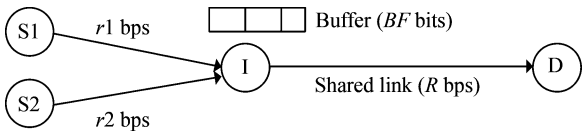

Fig. 1. Topology of experimental network.

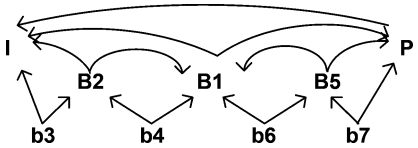

Fig. 2. Pyramid GOP structure; A B-frame in upper case can be used for reference while the ones in lower case cannot. The numbers indicate the coding order within the group.

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 [52]. 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 our factor consideration.

The Hint Track method in [36] and [37] cannot 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 router. On the other hand, the packet dropping policy in [36] and [37] cannot 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 (developed by ITS [4]). The VQM metric was found to be better correlated with human perception than two competing metrics, DVQ (Digital Video Quality) and VSSIM, as shown in [5].

We simulate the experiment using NS-2 [40] for a network topology shown in Fig. 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 10 s 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 nine 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 nine 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. 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.

A. PQ Comparison With DT and CMSE

Table V shows the comparison results for different buffer sizes when the bottleneck rate is fixed at $\mathrm{R} = 1200$ 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 $(\mathrm{R} = 800$ kbps$)$ and the results can be seen in Table VI. We observe a similar trend as in Table V (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 V (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. However, the model is still capable of prioritizing the video well and outperforms other policies.

Table VII 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 $(\mathrm{R} = 1200$ kbps$)$ than at lower rates. Nevertheless, the performance of our model is quite robust

TABLE V
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

Pyramid (R=1200)				**IBBP** (R=1200)				**IPPP** (R=1200)			
vs. **DT** (comparison ratio =1.27)	Wins	Losses	Ties	vs. **DT** (comparison ratio =2)	Wins	Losses	Ties	vs. **DT** (comparison ratio =5)	Wins	Losses	Ties
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** (comparison ratio =6.14)	Wins	Losses	Ties	vs. **cMSE** (comparison ratio =5)	Wins	Losses	Ties	vs. **cMSE** (comparison ratio =0.68)	Wins	Losses	Ties
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

TABLE VI
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

Pyramid (R=800)				**IBBP** (R=800)				**IPPP** (R=800)			
vs. **DT** (comparison ratio =1.21)	Wins	Losses	Ties	vs. **DT** (comparison ratio =1.57)	Wins	Losses	Ties	vs. **DT** (comparison ratio =3.5)	Wins	Losses	Ties
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** (comparison ratio =2.5)	Wins	Losses	Ties	vs. **cMSE** (comparison ratio =2.85)	Wins	Losses	Ties	vs. **cMSE** (comparison ratio =0.459)	Wins	Losses	Ties
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

TABLE VII
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

Pyramid (BF=300)				**IBBP** (BF=80)				**IPPP** (BF=80)			
vs. **DT** (comparison ratio =1.07)	Wins	Losses	Ties	vs. **DT** (comparison ratio =1.57)	Wins	Losses	Ties	vs. **DT** (comparison ratio =4.4)	Wins	Losses	Ties
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** (comparison ratio =2.17)	Wins	Losses	Ties	vs. **cMSE** (comparison ratio =3.9)	Wins	Losses	Ties	vs. **cMSE** (comparison ratio =0.74)	Wins	Losses	Ties
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

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 VIII 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 VII and 2.39 for Table VIII). This shows that we consistently perform better than other policies across different fixed buffer sizes.

From Tables V–VIII, 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,

TABLE VIII
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

Pyramid (BF=600)				**IBBP** (BF=140)				**IPPP** (BF=140)			
vs. **DT** (comparison ratio io=1.5)	Wins	Losses	Ties	vs. **DT** (comparison ratio =2.6)	Wins	Losses	Ties	vs. **DT** (comparison ratio =3)	Wins	Losses	Ties
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** (comparison ratio =3.28)	Wins	Losses	Ties	vs. **cMSE** (comparison ratio =3.5)	Wins	Losses	Ties	vs. **cMSE** (comparison ratio =0.5)	Wins	Losses	Ties
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

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 our 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 our model is the same for all frames and cannot 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, our visibility model just uses initial MSE (MSE in the frame where the packet loss occurred) which is computationally trivial.

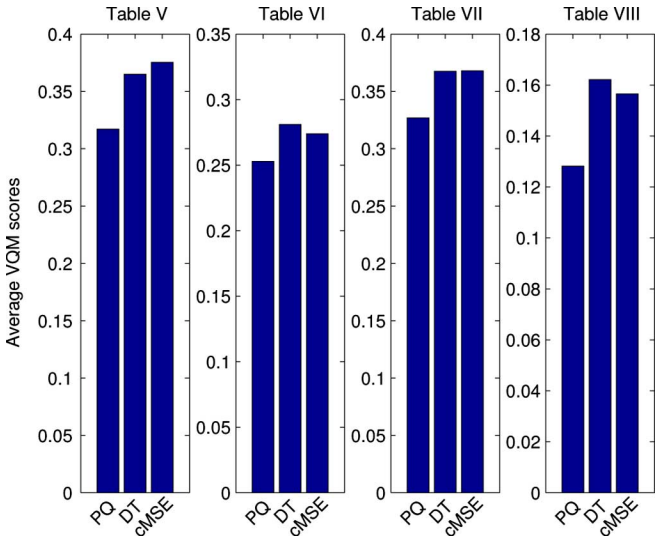

Fig. 3. Comparisons of average VQM scores among PQ, DT, and cMSE over the competitions in Tables V–VIII.

Another performance comparison is illustrated in Fig. 3, where average VQM scores among PQ, DT, and cMSE over the competitions in Tables V–VIII are shown. We can see that on average, the VQM scores obtained by PQ are lower (better) than that 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 V–VIII, but also on average has lower (better) VQM scores over different comparisons.

Although our proposed visibility model is built using data of isolated losses (one packet loss for every 4 s, as discussed in Section 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. Our model does well consistently across different buffer sizes and transmission rates. Note

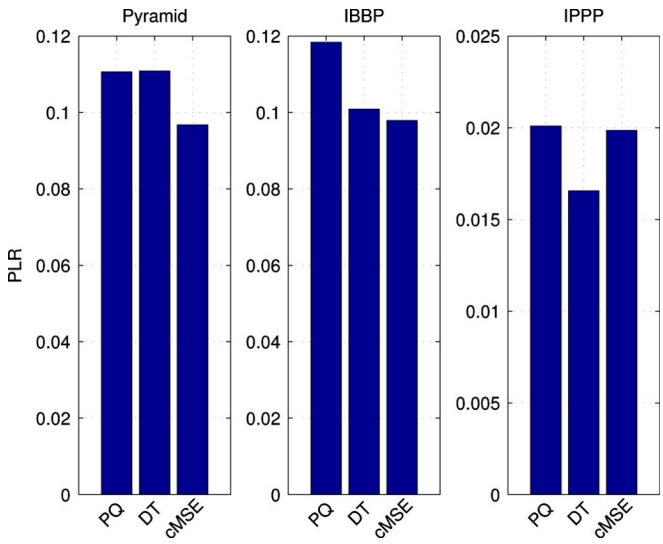

Fig. 4. Average PLR (Packet Loss Rate) comparisons among PQ, DT, and cMSE in different GOP structures.

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., [16] and [18]).

B. Packet Loss Rate for PQ, CMSE, and DT

In addition to VQM score comparisons for various network conditions detailed in Section IV-A, 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 Fig. 4. A PLR value corresponding to a dropping method in a GOP is obtained by averaging the PLRs from corresponding (R, BF) pairs listed in the tables from Section IV-A. Fig. 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 Section IV-A. 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 as discussed in Section I may not relate well to perceptual video quality.

VII. Conclusion

In this paper, we propose a generalized linear model for packet loss visibility applicable to different GOP structures and a perceptual-quality based packet dropping policy for a router to intelligently drop packets, when necessary, to minimize the degradation in visual quality. The contributions of this paper are the following: (a) 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. (b) We use our visibility model to prioritize video packets and design a policy for perceptual-quality based packet discarding. Experiments done under diverse network conditions and GOP structures show that the proposed PQ policy 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. (c) 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.

References

[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans . Image Process.*, vol. 13, no. 4, Apr. 2004.

[2] Z. Wang, L. Lu, and A. C. Bovik, "Video quality assessment based on structural distortion measurement," *Signal Process.g: Image Commun.*, vol. 19, no. 2, Feb. 2004, Special Issue on Objective Video Quality Metrics.

[3] M. A. Masry and S. S. Hemami, "A metric for continuous quality evaluation of compressed video with severe distortions," *Signal Process.: Image Commun.*, vol. 19, no. 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," presented at the IEEE Int. Conf. Image Processing, Oct. 2006.

[6] S. Wolf and M. H. Pinson, "Low bandwidth reduced reference video quality monitoring system," presented at the 1st Int. 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 Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 71–83, Jan. 2008.

[8] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "Perceptual blur and ringing metrics: Application to Jpeg2000," *Signal Process.: Image Commun.*, pp. 163–172, Feb. 2004.

[9] A. Eden, "No-Reference Estimation of the Coding PSNR for H.264-Coded Sequences," *IEEE Trans. Consum. Electron.*, vol. 53, no. 5, pp. 667–674, May 2007.

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

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

[12] S. Tao, J. Apostolopoulos, and R. A. Guerin, "Real-time monitoring of video quality in ip networks," *Proc. NOSSDAV*, pp. 129–134, Jun. 2005.

[13] A. R. Reibman, V. Vaishampayan, and Y. Sermadevi, "Quality monitoring of video over a packet network," *IEEE Trans. Multimedia*, vol. 6, no. 4, pp. 327–334, Apr. 2004.

[14] G. W. Cermak, Videoconferencing Service Quality as a Function of Bandwidth, Latency, and Packet Loss T1A1.3/2003-026, Verizon Laboratories, 2003.

[15] B. Chen and J. Francis, "Multimedia Performance Evaluation," *AT&T Tech. Mem.*, Feb. 2003.

[16] S. Mohamed and G. Rubino, "A study of real-time packet video quality using random neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1071–1083, Dec. 2002.

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

[18] R. V. Babu, A. S. Bopardikar, A. Perkis, and O. I. Hillestad, "No-Reference metrics for video streaming applications," presented at the Int. Workshop on Packet Video, Dec. 2004.
[19] H. Rui, C. Li, and S. Qiu, "Evaluation of packet loss impairment on streaming video," *J. Zhejiang Univ. SCIENCE*, vol. 7, Apr. 2006.
[20] S. Winkler and R. Campos, "Video quality evaluation for internet streaming applications," *SPIE, Human Vis. Electron. Imag. VIII*, vol. 5007, pp. 104–115, Jan. 2003.
[21] R. R. Pastrana-Vidal and J.-C. Gicquel, "Automatic quality assessment of video fluidity impairments using a no-reference metric," presented at the Int. Workshop on Video Processing and Quality Metrics, Jan. 2006.
[22] A. R. Reibman, S. Kanumuri, V. Vaishampayan, and P. C. Cosman, "Visibility of individual packet losses in MPEG-2 video," presented at the IEEE Int. Conf. Image Processing, Oct. 2004.
[23] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Pacific Grove, CA: Wadsworth, 1984.
[24] S. Kanumuri, P. C. Cosman, and A. R. Reibman, "A generalized linear model for MPEG-2 packet-loss visibility," presented at the International Packet Video Workshop, Dec. 2004.
[25] S. Kanumuri, P. C. Cosman, A. R. Reibman, and V. Vaishampayan, "Modeling packet-loss visibility in MPEG-2 video," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 341–355, Apr. 2006.
[26] P. McCullagh and J. A. Nelder, *Generalized Linear Models*, 2nd ed. New York: Chapman & Hall, 1989.
[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," presented at the IEEE Int. Conf. Image Processing, Oct. 2006.
[28] A. R. Reibman and D. Poole, "Characterizing packet loss impairments in compressed video," presented at the IEEE Int. Conf. Image Processing, Sep. 2007.
[29] A. R. Reibman and D. Poole, "Predicting packet-loss visibility using scene characteristics," in *Proc. Int. Packet Video Workshop*, Sep. 2007, pp. 308–317.
[30] J. H. Friedman and N. I. Fisher, "Bump hunting in high-dimensional data," *Statist. Comput.*, vol. 9, pp. 123–143, 1999.
[31] J. C. De Martin and D. Quaglia, "Distortion-based packet marking for MPEG video transmission over diffserv networks," in *Proc. ICME*, Oct. 2001, pp. 111–116.
[32] F. De Vito, L. Farinetti, and J. C. De Martin, "Perceptual classification of MPEG video for differentiated-services communications," in *Proc. ICME*, Aug. 2002, vol. 1, pp. 141–144.
[33] D. Quaglia and J. C. De Martin, "Adaptive packet classification for constant perceptual quality of service delivery of video streams over time-varying networks," in *Proc. ICME*, Jul. 2003, vol. 3, pp. 369–72.
[34] J. Chakareski, J. Apostolopoulos, and B. Girod, "Low-complexity rate-distortion optimized video streaming," presented at the Int. Conf. Image Processing, 2004.
[35] J. Chakareski, J. G. Apostolopoulos, S. Wee, and B. Girod, "Rate-distortion hint tracks for adaptive video streaming," *IEEE Trans Circuits Syst. Video Technol.*, vol. 15, 2005.
[36] J. Chakareski and P. Frossard, "Rate-distortion optimized bandwidth adaptation for distributed media delivery," presented at the IEEE Int. Conf. Multimedia and Expo, 2005.
[37] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Trans. Multimedia*, vol. 8, pp. 207–218, Apr. 2006.
[38] W. Tu, W. Kellerer, and E. Steinbach, "Rate-distortion optimized video frame dropping on active network nodes," *Packet Video*, 2004.
[39] W. Tu, J. Chakareski, and E. Steinbach, "Rate-distortion optimized frame dropping and scheduling for multi-user conversational and streaming video," *J. Zhejiang Univ.-Sci. A*, 2006.
[40] NS Project [Online]. Available: http://www.isi.edu/nsnam/ns/
[41] T.-L. Lin, Y. Zhi, S. Kanumuri, P. Cosman, and A. Reibman, "Perceptual quality based packet dropping for generalized video GOP structures," presented at the Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP), 2009.
[42] Y. Sermadevi and A. R. Reibman, Unpublished Subjective Test Results, 2002.
[43] O. Nemethova, M. Ries, M. Zavodsky, and M. Rupp, "PSNR-based estimation of subjective time-variant video quality for mobiles," presented at the MESAQUIN, 2006.
[44] A. Hanjalic, *Content-Based Analysis of Digital Video*. Boston, MA: Kluwer, 2004.
[45] Z. Liu, D. Gibbon, E. Zavesky, B. Shahraray, and P. Haffner, AT&T Research at TRECVID, 2006.
[46] 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 Trans. Circuits Syst. Video Technol.*, vol. 10, no. 2, pp. 133–146, Feb. 2000.
[47] K. Stuhlmuller, N. Farber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.
[48] M. Budagavi and J. D. Gibson, "Multiframe video coding for improved performance over wireless channels," *IEEE Trans. Image Process.*, vol. 10, no. 2, pp. 252–265, Feb. 2001.
[49] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, 2nd ed. New York: Wiley-Interscience, 2000.
[50] R Project [Online]. Available: http://www.r-project.org/
[51] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York: Springer-Verlag, 2001.
[52] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, 3rd ed. Reading, MA: Addison Wesley, 2004.

Ting-Lan Lin (S'08) received the B.S. and M.S. degrees in electronic engineering from Chung Yuan Christian University, Chung Li, Taiwan, R.O.C., in 2001 and 2003, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering at the University of California at San Diego, La Jolla.

During the summer of 2008, he interned in the Display System group at Qualcomm, San Diego, CA. His research interests include video compression, video streaming in lossy networks, optimization of packet prioritization, and perceptual video quality.

Sandeep Kanumuri (S'04–M'07) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Madras, India, in 2002, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at San Diego (UCSD), La Jolla, in 2004 and 2006, respectively.

He is currently a Research Engineer at DoCoMo USA Labs. During the summer of 2003, he was a Visiting Researcher at AT&T Labs, Florham Park, NJ. During the summer of 2005, he worked in the MediaFLO division of Qualcomm, San Diego, CA.

Dr. Kanumuri received the DoCoMo Communications Laboratories President Award and the CAL-IT2 Fellowship. He works on research problems related to the compression, distribution, postprocessing and quality estimation of images and video.

Yuan Zhi received the B.S. degree in electrical engineering and the B.A. degree in english language and literature from the University of Maryland, College Park, in 2007, and the M.S. degree in electrical engineering from the University of California at San Diego, La Jolla, in 2009.

She is currently with Texas Instruments, Houston, TX.

David Poole is a Principal Member of Technical Staff in the Statistics Department, AT&T Labs-Research, Florham Park, NJ. He is the Program Chair-elect for the Section on Statistical Computing of the American Statistical Association. He has extensive experience with large-scale data mining algorithms, computer-intensive statistical procedures, traffic engineering, and fraud detection.

Pamela C. Cosman (S'88–M'93–SM'00–F'08) received the B.S. degree with Honors in electrical engineering from the California Institute of Technology, Pasadena, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1989 and 1993, respectively.

She was an NSF postdoctoral fellow at Stanford University and a Visiting Professor at the University of Minnesota during 1993–1995. In 1995, she joined the faculty of the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, where she is currently a Professor. She was the Director of the Center for Wireless Communications from 2006 to 2008. Her research interests are in the areas of image and video compression and processing and wireless communications.

Dr. Cosman is the recipient of the ECE Departmental Graduate Teaching Award (1996), a Career Award from the National Science Foundation (1996–1999), a Powell Faculty Fellowship (1997–1998), and a Globecom 2008 Best Paper Award. She was a guest editor of the June 2000 special issue of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS on "Error-resilient image and video coding," and was the Technical Program Chair of the 1998 Information Theory Workshop in San Diego. She was an associate editor of the IEEE COMMUNICATIONS LETTERS (1998–2001), and an associate editor of the IEEE SIGNAL PROCESSING LETTERS (2001–2005). She was a senior editor (2003–2005) and is now the Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. She is a member of Tau Beta Pi and Sigma Xi.

Amy R. Reibman received the B.S., M.S., and Ph.D. degrees in electrical engineering from Duke University, Durham, NC, in 1983, 1984, and 1987, respectively.

From 1988 to 1991, she was an Assistant Professor in the Department of Electrical Engineering at Princeton University, Princeton, NJ. In 1991, she joined AT&T Bell Laboratories and became a Distinguished Member of Technical Staff in 1995. She is currently a Lead Member of Technical Staff in the Communication Sciences and Artificial Intelligence Research Department, AT&T Laboratories, Florham Park, NJ. Her research interests include video compression systems for transport over packet and wireless networks, video quality metrics, superresolution image and video enhancement, and 3-D and multiview video.

Dr. Reibman was elected IEEE Fellow in 2005 for her contributions to video transport over networks. In 1998, she won the IEEE Communications Society Leonard G. Abraham Prize Paper Award. She was the Technical Co-Chair of the IEEE International Conference on Image Processing in 2002; the Technical Co-Chair for the First IEEE Workshop on Multimedia Signal Processing in 1997; and the Technical Chair for the Sixth International Workshop on Packet Video in 1994. She was a Distinguished Lecturer for the IEEE Signal Processing Society from 2008–2009.