A Generalized Linear Model for MPEG-2 Packet-Loss Visibility

Sandeep Kanumuri
Univ. Calif. at San Diego
skanumur@code.ucsd.edu

Pamela C. Cosman
Univ. Calif. at San Diego
pcosman@code.ucsd.edu

Amy R. Reibman
AT&T Labs – Research
amy@research.att.com

Abstract— **In this paper, we focus on predicting the visibility of packet losses in MPEG-2 compressed video streams. We develop a generalized linear model (GLM) to predict the probability that a packet loss will be visible to an average viewer. The GLM input consists of parameters that can be easily extracted from the video near the location of the loss, and outputs an estimate of the probability that that loss is visible. We also show how our GLM can be used to classify each loss as visible or invisible. Using this method, we are able to achieve a high classification accuracy.**

I. Introduction

When sending compressed video across today's communication networks, packet losses may occur. Network service providers would like to (a) provision their network to keep the packet loss rate below an acceptable level, and (b) monitor the traffic on their network to assure continued acceptable video quality. Unfortunately, each packet loss in video has a different visual impact. For example, one may last for a single frame while another may last for many; one may occur in the midst of an active scene while another is in a motionless area. Thus, the problem of evaluating video quality given packet losses is challenging.

In this paper, we focus on predicting the visibility of packet losses in MPEG-2 compressed video streams. Our goal is to develop a quality monitor that is accurate, real-time, can operate on every stream in the network, and answers the question, "How are the losses present in this particular stream impacting its visual quality?". Toward this goal, we develop a generalized linear model (GLM) to predict the probability that a packet loss will be visible to an average viewer. The GLM input consists of parameters that can be easily extracted from the video near the location of the loss, and outputs an estimate of the probability that that loss is visible. We show how our GLM can be used to classify each loss as visible or invisible.

A lot of research has been done on developing objective perceptual metrics for compressed video not affected by network losses. While these metrics can predict the quality degradation caused due to compression artifacts, they are not equipped to handle the degradation caused by network losses.

In earlier efforts to understand the visual impact of packet losses [3], [4], [5], [6], the goal was to understand the average quality of typical videos subjected to average packet loss rates (PLR). Video conferencing is studied in [3] using the average judgement of consumer observers to examine the relative importance of bandwidth, latency, and packet loss. The impact of packet loss on the Mean Opinion Score (MOS) of real-time streaming media was studied in [4] for Microsoft Windows Media encoder 9 (beta version) video. A neural network was trained in [5] to viewer responses on the ITU-R 9-point quality scale, when a single 10-second sequence was subjected to different bandwidth, frame-rate, packet loss rate, and I-block refresh rate.

Hughes et al. [6] use MOS to evaluate the subjective quality of VBR video subjected to ATM cell loss over a 10-second period. They show that performance is sensitive not only to the magnitude of the bursts, but also to their frequency. "Very different" results were obtained for different sequences. Other challenges identified by these authors [6] were: (a) many different realizations of both packet loss and video content are necessary to reduce the variability of viewer responses; (b) very low PLRs are difficult to explore because the typical test period (10 seconds) is so short that typical realizations may have no packet losses; (c) the "forgiveness effect" causes viewers to rank a long video based on more recently viewed information. The joint impact of encoding rate and ATM cell losses on MPEG-2 video quality was studied in [13]. Here the quality of video is judged based on an existing perceptual quality metric and not based on subjective tests. A framework for employing objective perceptual quality assessment methods, evaluating the quality of audio, video and multimedia signals, to model network performance is demonstrated in [15].

In addition, these studies [3], [4], [5], [6] all use MOS to evaluate quality. However, the MOS quality rating methodology has a number of difficulties, as detailed in [7]. First, the impairment (or quality) scales are generally not interpreted by subjects as having equal step-size, and labels in different languages are interpreted differently. Second, subjects tend to avoid the end-points of the scales. Third, the term "quality" itself is actually not a single variable, but has many dimensions.

Thus, we designed and conducted a subjective test that does not use MOS, and explores the impact of each packet loss individually. Viewers are shown MPEG-2 video with injected packet losses, and asked to indicate when they see an artifact in the displayed video. Data is gathered for a total of 1080 packet losses over 72 minutes of MPEG-2 video. "Ground truth" for the probability of visibility of packet losses is defined by the results of our subjective tests. The frequency of visible

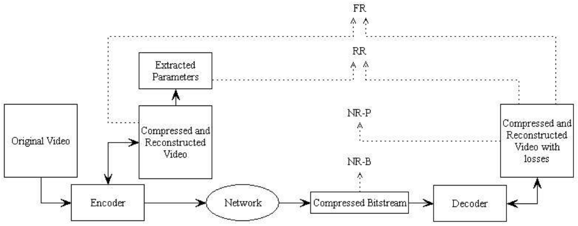

Fig. 1. Illustration of FR, RR and NR methods

packet losses will have a significant influence on the perceived quality; however, in this study, we don't explore this issue.

In our previous work [1], we designed a classifier to classify each packet loss as visible or invisible to an average human observer. Our classifier was a tree where the path at each node is based on a binary decision using one factor that affects visibility. Using this classifier, we were unable to differentiate packet losses which are at the threshold of visibility from those far away from the threshold. In this paper, because we would like to predict the probability that a packet loss will cause a visible artifact, we are motivated to use a GLM instead of a decision tree. We also explore the different factors that affect packet-loss visibility.

Figure 1 illustrates different methods for quality assessment based on locations for measuring networked video. Full-Reference (FR) methods are based on measurements of the exact pixel values at both the encoder and decoder. Reduced-Reference (RR) methods are based on measurements of certain key parameters at the encoder and access to the exact pixel values at the decoder. No-Reference (NR) methods do not have access to any measurements at the encoder. There are two types of NR methods: NR-pixel (NR-P) and NR-Bitstream (NR-B) methods. NR-P methods can measure the decoded video at the pixel level, while NR-B methods can measure only the bitstream, not the decoded pixels. FR methods might give the highest accuracy, but NR-B methods are the best choice for network-based quality monitoring. They can be deployed at different points in the network without the additional complexity of a decoder for every stream. Gastaldo et al. [14] used a Neural Network approach to design an objective quality assessment algorithm for MPEG-2 video streams without decoding. However, the algorithm is based on compression artifacts and does not consider network losses like loss of packets.

In [1], our classifier was based on factors that depended on a complete video bitstream, the location of the loss in the received bitstream, and the complete decoded video. As a result, this classifer must be considered to be a RR quality metric. In this paper, we explore a range of quality metrics (all using the GLM structure), which differ in the amount of information that is available at the time of measurement. Specifically, we consider an RR, an NR-P, and an NR-B method, all based on parameters easily extracted from available bitstreams, and explore the relative quality of each.

This paper is organized as follows. Section II gives an overview of MPEG-2 packet losses and their impact. Section III describes our subjective test. Section IV describes the logistic regression model, the GLM which suits our purpose. Section V describes the objective factors that we believe should be included in our models. Section VI describes our statistical analysis and its results, while Section VII concludes.

II. Effect of a Packet Loss

MPEG-2 is typically packetized in one of two ways. First, video can be segmented and packetized into small fixed-size packets (like ATM cells or MPEG-2 Transport Stream packets), in which case a single packet loss might force the decoder to discard either a slice or an entire frame. Second, a variable-sized packet can contain one or more slices. In both cases, a packet loss corresponds to the loss of one or more slices. We explore here the case that a packet loss causes the loss of a single slice, a double slice, or the entire frame.

The initial error caused by a packet loss propagates in space and time as a result of the video decoding algorithm. The exact error due to packet loss can be completely described by (a) the initial error for each macroblock in the lost packet, and (b) the macroblock type and (c) motion information for subsequently received macroblocks [11]. The latter two control the temporal duration and spatial spread of the error.

The initial error induced by a packet loss depends on the error concealment strategy used by the decoder. A typical concealment strategy, used here, is zero-motion concealment, in which an affected macroblock is estimated using the macroblock in the same spatial location from the closest reference frame. In this case, the initial error is simply the difference between the current encoded frame and the closest reference frame for the affected macroblocks.

We expect the visibility of a loss to depend on a complex interaction of its location, the video encoding parameters, and the underlying characteristics of the video signal itself. For example, the texture and motion of the underlying signal may potentially mask the error. To isolate the impact of the various parameters, one approach could be to inject different error amplitudes against an identical signal background, as was done in [12] for blocky, blurry and noisy artifacts. However, for packet losses, the error itself is highly dependent on the underlying signal and so we do not have control over the amplitude of the error. Therefore, we must take a different approach.

When choosing the packet losses to inject for our subjective tests, we have independent control over the location, initial spatial extent and temporal duration of each loss we inject. The other factors depend on the signal. Thus, we choose whether to lose a single slice, double slice or an entire frame. We also choose the loss to be in a B-frame (which would last a single frame) or in a reference frame (which will last until the next I-frame). In choosing the location of the loss, we distribute the locations vertically within the frame and choose representative samplings from both still and active regions of the sequence.

III. Subjective Tests

For the subjective tests, we can conduct either a single-stimulus test or a double-stimulus test. In a single-stimulus test, only the video being evaluated (here, video with packet losses) is shown. The reference or original video is not shown. In a double-stimulus test, both videos are shown. We conducted a single-stimulus test because the 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. The viewer bases his/her judgement on the lossy video only.

In the test, the viewers' task is to indicate when they saw an artifact, where an artifact is defined simply as a glitch or abnormality. We wanted viewers to be immersed in the viewing process and not scrutinizing the video for any possible impairment. Thus we chose DVD-quality MPEG-2 video[1] from travel documentaries. Audio was not presented. Zero-motion error concealment using the closest reference frame was used whenever there was a packet loss. This presumes a minimum amount of intelligence on the part of the decoder. Decoders that use sophisticated error concealment methods may have fewer visible packet losses. However, since we would like to predict the visibility of packet losses in the network, without necessarily knowing which decoder the viewer is using, we assume only this minimal error concealment strategy.

We chose twelve 6-minute video sequences, for a combined length of 72 minutes. We grouped the sequences into 4 sets, each consisting of three sequences. This limited a viewing session to 18 minutes so as not to tire or bore the viewers. During each session, a viewer evaluated a set of video sequences with a short break between each sequence. Some viewers participated in more than one viewing session, although never on the same day. Each set of video sequences (and hence each packet loss) was evaluated by 12 viewers.

Viewers were told that the videos they were watching would have impairments caused by packet losses, and that when they saw something unexpected in the video like a glitch, they should respond by pressing the space bar. They were asked to keep their finger on the space bar so they would not be distracted by that task. The lighting condition was typical of an office environment and the viewer was positioned approximately six picture heights from the screen.

A total of 1080 packet losses were randomly injected in these videos such that every non-overlapping four-second interval contained one packet loss in the first three seconds. The one-second guard interval ensured a viewer had sufficient time to respond to each individual error. We distributed the losses such that 30% affected an entire frame, 10% affected two adjacent slices, and 60% affected a single slice. Further, we chose to have 30% of the losses to be in B-frames (and hence have a temporal duration of one frame), and the remaining 70% evenly distributed across the available P- and I-frames in the 3-second interval. Finally, the video we selected was highly varied, with many different motion types and amounts of spatial texture. Therefore, we believe our injected packet losses occur across a representative set of diverse signal background types.

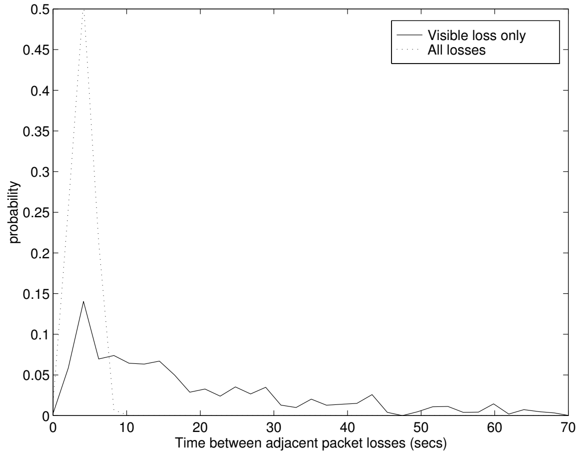

Fig. 2. Histogram of time between adjacent packet losses

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. We processed these to create a matrix with 1080 rows and 12 columns, whose entries indicate whether a viewer responded to a packet loss or not. If a viewer pressed the space bar with in two seconds after a packet loss occurred, he/she is considered to have responded to that packet loss. Otherwise, he/she is considered not to have responded to the packet loss. The ground truth for the probabilities of visibility of a packet loss was defined from these viewers' responses. The probabilities were calculated as the number of viewers who saw the packet loss divided by 12.

Viewers were not told the pattern of injected packet losses. There is a concern, however, that while viewing the video they might infer that a packet loss occurs in every 4-second interval. If a viewer were able to predict this, it might bias their responses. To analyze this, we examined the time between adjacent packet losses, and time between adjacent *visible* packet losses, where we define visible to be those losses that over 75% of viewers indicated they saw. Figure 2 shows that the density of the time between adjacent packet losses is triangular with a minimum, mean, and maximum of one, four, and seven seconds, as expected. Also shown in Figure 2 is the density of time between adjacent visible packet losses, with its long tail out to 130 seconds not shown. This density has a peak near four seconds. However, fewer than 5% of all the losses are visible within 7 seconds of each other. This means that viewers saw adjacent packet losses less than 5% of the time. Therefore, we do not believe that viewers were able to infer that packet losses might occur in every four-second interval and begin to anticipate an artifact.

[1] 720 pixels, 480 lines, and 60 fields per second.

IV. GLM - Logistic Regression

In this paper, we model the probability of visibility using a Generalized Linear Model (GLM). Logistic Regression is a type of GLM which models the parameter p of a binomial distribution. Generalized linear models are an extension of classical linear models [2]. First we will give a brief overview of the classical regression problem and then explain the generalized linear model and logistic regression.

Let $y_1, y_2, ..., y_N$ be a realization of independent random variables $Y_1, Y_2, ..., Y_N$ such that Y_i has binomial distribution with index m_i and 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. We are trying to model the parameter p 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. An ordinary linear model between $\mathbf{p}$ and $\mathbf{X}$ can be written as

$$\mathbf{p} = \gamma + \sum_{j=1}^{P} \mathbf{x_j} \beta_j$$

where $\mathbf{x_j}$ is the j^{th} column of $\mathbf{X}$ and $\beta_1, \beta_2, ..., \beta_P$ are the coefficients of the factors. Coefficients β and the constant term γ are usually unknown and need to be estimated from the data. A simple linear regression model is incapable of estimating the parameter p of a binomial model because the output of a linear model typically has the range $(-\infty, \infty)$ while we know $p \in [0, 1]$.

A generalized linear model can be represented as

$$g(\mathbf{p}) = \gamma + \sum_{j=1}^{P} \mathbf{x_j} \beta_j$$

where $g(.)$ is called the link function, which is typically non-linear. Classical regression is a special case of GLM where the link function $g(.)$ is an identity. For logistic regression, the link function is the logit function, which is the canonical (therefore default) link function for the binomial distribution. The purpose of the link function here is to map $p \in [0, 1]$ onto the entire real line $(-\infty, \infty)$. The logit function is defined as

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

Given N observations, we can fit models using up to N parameters. The simplest model, also called the Null model, has only one parameter: the constant γ. At the other extreme, it is possible to have a model with as many parameters as there are observations, called the Full Model; however, this is problematic. The goodness of fit for generalized linear models can be characterized by the deviance value, which is formed as the logarithm of a ratio of likelihoods.

If we denote the log-likelihood function for model $\mathbf{p}$ (which is a function of β), and the observations $\mathbf{y}$ as $l(\mathbf{p}; \mathbf{y})$, then for the binomial distribution we can write the log-likelihood function as

$$l(\mathbf{p}; \mathbf{y}) = \sum_{i=1}^{N} [y_i log(\frac{p_i}{1-p_i}) + m_i log(1-p_i) + log(\begin{pmatrix} m_i \\ y_i \end{pmatrix})]$$

where m_i represents the number of trials made for observation i. The log-likelihood function $l(\mathbf{p}; \mathbf{y})$ is maximized for the full model. Let the full model and the current model be represented by $\tilde{\mathbf{p}}$ and $\hat{\mathbf{p}}$ respectively. Then, we can write $\tilde{p_i} = \frac{y_i}{m_i}$. Further, the deviance for the model represented by $\hat{\mathbf{p}}$ is defined as

$$D(\mathbf{y}; \hat{\mathbf{p}}) = 2[l(\tilde{\mathbf{p}}; \mathbf{y}) - l(\hat{\mathbf{p}}; \mathbf{y})].$$

From the definition, we can see that the deviance for the Full model is zero and the deviance for all other models is positive. So the smaller the deviance, the better the model fit. The deviance for the null model is also called the null deviance. The deviance is often used as a goodness-of-fit statistic for testing the adequacy of a fitted model. Under the assumptions of independence and $p \in (0, 1)$, the deviance can be shown to be asymptotically distributed as $\chi^2_{n-(P+1)}$, where (P+1) is the total number of parameters fitted for the model [2]. Furthermore, the difference in deviance between two models is also known to be approximately distributed as χ^2_k under the assumption of independence alone for large values of N, where k is the difference in the number of parameters estimated for each model. This is very useful in determining the significance of different factors.

We use the statistical software R [16] for our model fitting and analysis. To obtain the model parameters, R uses an iteratively re-weighted least-squares technique to generate a maximum-likelihood estimate. After fitting a particular model, the importance of each factor in the model can be evaluated by the resultant increase in deviance when we remove that factor from the model. This increase can be compared with the appropriate χ^2 statistic to compute the p-value for this factor. If the p-value is less than 0.05, then the factor is significant at the 95% level. We represent the observed probability of visibility as $\tilde{p}$ and the predicted probability of visibility as $\hat{p}$.

V. Factors affecting Visibility

In this section, we describe the objective parameters that we believe will be useful to model the probability of visibility. We focus primarily on factors that are easily extracted from the video, as our goal is to develop an NR-B method for evaluating video quality within a network. In the next section, we will explore the usefulness of these factors in our models.

These objective factors can be classified into two types: content-independent factors and content-specific factors. Content-independent factors depend on the location of the packet loss in the MPEG-2 bitstream, but do not depend on the content of the video. Content-independent factors can therefore be calculated exactly from the lossy bitstream itself. Content-specific factors depend on the content of the video at the location of the packet loss. Content-specific factors

Factor Acronym	Description
TMDR	Time Duration: Number of frames affected by the packet loss
SPTXNT	Spatial Extent: Number of slices lost
HGT	Height: Number of the topmost slice lost
MOTX	Average motion in x-direction
MOTY	Average motion in y-direction
VARMX	Variance of motion in x-direction
VARMY	Variance of motion in y-direction
RSENGY	Average residual energy per pixel after motion compensation
IMSE	Mean square error per lost pixel

TABLE I

DESCRIPTION OF FACTORS AFFECTING VISIBILITY

can be calculated exactly at the encoder side, by using the original bitstream without losses. However, these content-specific factors cannot be exactly obtained from a bitstream in which packets are already lost.

The content-independent factors we consider are Temporal Duration (TMDR), Spatial Extent (SPTXNT) and the vertical position (HGT) of the error induced by the packet loss. Temporal duration represents the number of frames that are affected by a packet loss, and varies from 1 to 13 in our bitstreams. An error in a B-frame will last a single frame, while an error in a reference frame will last until the next I-frame. The spatial extent represents the number of slices affected by the packet loss. In our case, it is either 1, 2 or 30 corresponding to a single slice, double slice or a frame loss. HGT is the number of the topmost slice affected by the packet loss, where the slices are numbered from 0 to 29 from top to bottom. This factor captures the varying attention viewers have on different regions in the frame. All the content-independent factors can be controlled at the time of choosing which losses to introduce. Since the content-independent factors can be extracted exactly from the lossy bitstream, they are identical across our RR, NR-P, and NR-B models.

Content-specific factors include Motion (MOTX and MOTY), Variance of motion (VARMX and VARMY), Residual Energy (RSENGY) and Initial Mean Square Error (IMSE). MOTX and MOTY represent the average motion in x and y directions respectively for the lost slices. VARMX and VARMY represent the variance of motion in x and y directions for the lost slices. RSENGY denotes the average residual energy per pixel after motion compensation in the lost slices. IMSE is the mean squared error per pixel between the decoded videos with and without packet loss evaluated only over the pixels in lost slices. Table I summarizes the descriptions of all the factors along with their acronyms.

The content-specific factors described above can be extracted exactly from the complete bitstream (available at the encoder) and the decoded pixels. Thus, they can only be exactly obtained using an RR method. Since the complete bitstream is not available to our NR-P and NR-B methods, they must estimate the content-specific factors for the missing slices. Further, to compute IMSE, decoded pixels are necessary; however, these are unavailable to the NR-B method.

For the RR method, the content-specific factors can be extracted for all slices, and this information can be made available to the quality monitor via reliable means. This information is then combined with the knowledge of which slices are lost to generate the set of parameters used in our RR models.

For the NR-P and NR-B methods, the parameters MOTX, MOTY, VARMX, VARMY and RSENGY are extracted directly from the bitstream for all *received* slices. Parameters for the missing slices are then estimated using one of two approaches. The first approach estimates the parameter using co-located slices in the previous frame. The second approach estimates the factor using spatially neighboring slices in the same frame. We tried each approach on one video sequence, to decide which approach performed best. For the MOTX, MOTY, VARMX, VARMY and RSENGY parameters, the first approach performed best for both the NR-P and NR-B cases.

For the NR-P case, the IMSE is computed for all received slices, where IMSE for received slices is defined to be the IMSE that would have resulted if the slice had been lost. The second approach above was found to be more effective for estimating the IMSE of the missing slices. For the NR-B method, neither of the above two approaches is possible since the decoded pixels aren't available. Thus, to estimate the IMSE for the NR-B case, we use the approach described in [11]. This approach [11] extracts and estimates additional parameters (like mean, spatial correlation, spatial variance) from the received slices to estimate IMSE for the missing slices.

VI. Statistical Analysis and Results

In this section, we apply Logistic Regression, a type of GLM, to the problem of estimating the probability that a packet loss is visible to an average viewer. We use the parameters extracted from our RR, NR-P, and NR-B methods to derive a separate model for each case. We explore a number of different sets of parameters, to determine the best way to characterize sequence motion and loss-duration for our objective.

We use the word "model" to characterize the set of parameters which comprise the matrix $\mathbf{X}$, introduced in section IV. We note that for each "model", we actually consider three: one for each of the RR, NR-P, and NR-B cases. The distinction between the three lies in whether the content-specific parameters are extracted exactly, or estimated as described in the previous section.

We begin with a model that uses the same factors as those in [1]. Next, we explore improved characterization of the motion variables. Finally, we explore improved characterization of the time-duration variables. Models in each subsection VI-A to VI-C are numbered using the corresponding subsection number (A-1, B-2 and C-3); intermediate models in a subsection have an additional qualifier. The Null Model is called Model 0.

A. Initial Model

Our initial model, Model 1, uses the factors TMDR, SPTXNT, MOTX, MOTY, VARMX, VARMY, RSENGY, IMSE and HGT for all three methods i.e. RR, NR-P and NR-B. In this model, we considered SPTXNT as a categorical variable with three levels to distinguish the three cases of single, double slice loss and frame loss errors. A categorical variable with N levels is treated as a vector of $N-1$ boolean variables. (The N-th level is represented by setting all $N-1$ boolean variables to zero.) For SPTXNT, we therefore considered two boolean variables: SPTXNT-2 and SPTXNT-30. SPTXNT-1 is considered default and its effect is included in the constant term.

All these factors formed the columns of $\mathbf{X}$. The null deviance obtained for our set of observations was 9254.8 with 1079 degrees of freedom for the χ^2 distribution. The deviances obtained with Model 1 for the RR, NR-P and NR-B cases were 5707.5, 6115.5 and 6114.8 respectively with 1069 degrees of freedom. This model has order 11: SPTXNT uses 2 degrees of freedom, the remaining 8 factors and the constant γ use 9 degrees of freedom. The MSE between $\tilde{p}$ from the full model and $\hat{p}$ from Model 1 is 0.0678 for RR, 0.0742 for NR-P and 0.0749 for NR-B case.

B. Improved Motion Variables

Next, we consider the effect of overall motion and its direction, rather than the effect of x and y directional motions. We define MOTM and MOTA to represent the magnitude and angle of average motion and VARM to represent the variance in average motion. We calculate MOTM, MOTA and VARM as follows:

$$MOTM = \sqrt{MOTX^2 + MOTY^2}$$

$$MOTA = \arctan(\frac{MOTY}{MOTX})$$

$$VARM = VARMX + VARMY$$

So our new model, Model 2a, consists of TMDR, SPTXNT, MOTM, MOTA, VARM, RSENGY, IMSE and HGT. The deviance values decrease with this model for the RR and NR-P cases, though the model order (degrees of freedom) is also reduced from 11 to 10. The deviance values obtained were 5670.6, 6107.6 and 6126 for the RR, NR-P and NR-B cases respectively, with 1070 degrees of freedom.

More importantly, we observed that the variable MOTA is not significant at the 95% level with its p-value being 0.1286, 0.2146 and 0.2556 for RR, NR-P and NR-B cases respectively. Thus, MOTA should be removed from the model. This decreases the model order by one, at the expense of a small increase in deviance. In the model without MOTA, Model 2b, the deviance values are trivially larger: 5672.9, 6109.2 and 6127.3 for the RR, NR-P and NR-B cases respectively, with 1071 degrees of freedom.

Our previous research [1] showed that packet losses are invisible when the overall motion is low. To account for this effect, we added a Boolean variable HIGHMOT which is set when $MOTM > 0.707$. This threshold was set to correspond to motion that is greater than half a pixel per frame in both x and y directions. This additional variable serves to allow the model to use a different constant value for high-motion slices as opposed to low-motion slices. Including this variable further reduces the deviance values by more than 350, which is highly significant. The new deviance values are 5175.4, 5698.2 and 5765.9 for the RR, NR-P and NR-B cases respectively, with 1070 degrees of freedom. We denote the final model of this subsection as Model 2. The MSE between $\tilde{p}$ from the full model and $\hat{p}$ from Model 2 is 0.0615 for RR, 0.0689 for NR-P and 0.0704 for NR-B case.

C. Improved Time-duration Variables

Our previous research [1] also showed that if $TMDR = 1$, which happens if the packet loss is in a B-frame, then the packet loss is almost always invisible. However, we also observed that the correlation coefficient between number of viewers who saw a packet loss and TMDR was 0.051, which is very low. This shows that instead of TMDR, the particular instance of $TMDR = 1$ has a significant effect on visibility. So instead of TMDR, we introduce a Boolean variable BFRAME which is set whenever a packet loss occurs in a B-frame. This modification to the model reduces the deviance very significantly in all the three cases. The new deviance values are 4939.1, 5323.2 and 5340.1 for the RR, NR-P and NR-B cases respectively, with 1070 degrees of freedom. This is Model 3a.

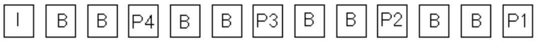

Fig. 3. FRAMETYPE value for different frames in a GOP

Next, we extend the boolean variable BFRAME to a categorical variable FRAMETYPE with 6 levels, depending on the type of frame in which the packet loss occurred. These 6 levels correspond to a B-frame, four P-frames with a different distance to the next I-frame, and an I-frame. We represent these types as B,P1,P2,P3,P4 and I. Figure 3 illustrates how these frames occur in the GOP structure of our videos. FRAMETYPE captures all the information in the temporal duration of a packet loss, and is due to the motion-compensated prediction in the decoder. For example, a packet loss in a P3 frame will have a temporal duration of 9. Including FRAMETYPE instead of BFRAME further reduced the deviance values to 4797.6, 5106.7 and 5115.7 for the RR, NR-P and NR-B cases respectively, with 1066 degrees of freedom. This is our final model, denoted as Model 3, which uses the factors FRAMETYPE, SPTXNT, MOTM, HIGHMOT, VARM, RSENGY, IMSE and HGT to predict the probability of visibility of a packet loss. The MSE obtained between $\tilde{p}$ and $\hat{p}$ from Model 3 is 0.0565 for RR, 0.0608 for NR-P, and 0.0611 for the NR-B case.

To verify the applicability of this model to new data, we perform a 4-fold cross-validation procedure. For this, we use

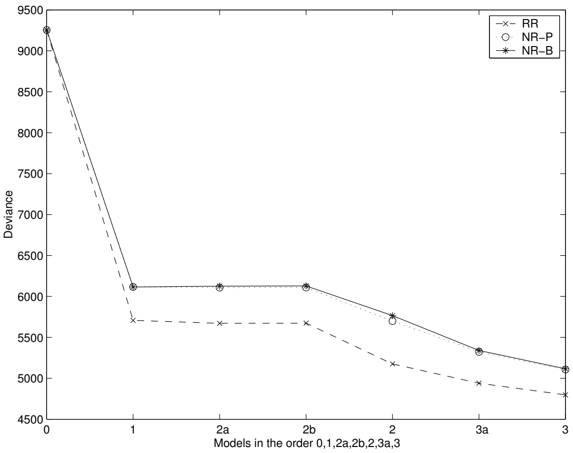

Fig. 4. Plot of Deviance for models considered

factor	coefficient
constant γ	-4.53
FRAMETYPE-P1	2.116
FRAMETYPE-P2	2.104
FRAMETYPE-P3	2.117
FRAMETYPE-P4	2.188
FRAMETYPE-I	5.326e-01
SPTXNT-2	7.161e-01
SPTXNT-3	1.54
MOTM	4.212e-01
HIGHMOT	1.398
VARM	-1.144e-02
RSENGY	-6.902e-03
IMSE	9.890e-04
HGT	-2.797e-02

TABLE II

COEFFICIENTS FOR MODEL 3 IN NR-B

Factor	Deviance increase		
	RR	NR-P	NR-B
FRAMETYPE	408.1	627.2	703.8
SPTXNT	532.4	517.4	440.3
MOTM	347.5	276	307
HIGHMOT	514.5	436.7	382.9
VARM	62.9	103	104
RSENGY	44.4	19.7	28
IMSE	439.8	197.9	188.9
HGT	38.7	47.9	56.6

TABLE III

FACTOR SIGNIFICANCE

the data from three out of the four sets of video as a training set. The data from the remaining set is used for testing. We repeat this process four times, each time choosing a different set for the testing set. Thus we have a predicted probability for each packet loss obtained when the packet loss was not used for training. The MSE obtained between $\tilde{p}$ and $\hat{p}$ during cross-validation for Model 3 is 0.0627 for RR, 0.065 for NR-P and 0.0647 for NR-B case. This shows that the model continues to perform well when encountering new data.

The improvement in models from the null model (Model 0) to the final model (Model 3) can be summarized by the plot of deviance, shown in figure 4, for all three cases (RR, NR-P and NR-B). There is a huge drop in deviance from the null model to the starting model (Model 1), which is expected. When we improve the treatment of the motion variables and also reduce the model order (Model 2), we see a decrease in deviance indicating a better fit. Also, we see a further decrease in deviance from Model 2 to Model 3 when we treat the time-duration information using a boolean structure.

The coefficients (γ and βs) for the final model (Model 3) in the NR-B case are tabulated in Table II. The values of the coefficients do not necessarily convey the importance of corresponding factors because these factors differ in their variances and in the range of values they take.

The significance of different factors in the model can be understood by the increase in the deviance that results if each factor is individually removed from the model. Table III shows the increase in deviance values for each factor, for the RR, NR-P and NR-B cases. From the table, we see that FRAMETYPE, SPTXNT, MOTM, HIGHMOT and IMSE are very significant factors affecting visibility. Since HIGHMOT depends completely on MOTM, we can attribute its importance also to MOTM. If we consider it this way, then MOTM becomes the most significant factor affecting visibility.

D. Classification Problem

Until now, we have considered the problem of predicting the probability of visibility. So we have considered a regression problem and modeled the probability using GLMs. In this subsection, we describe one way to use our model for classifying packet losses, and we analyze the results.

For this study, we classify a packet loss to be visible, invisible, or indeterminate, based on its probability of visibility. We divide the interval $[0, 1]$ into three regions, using the parameter α:

[0,0.5-α]	Invisible region
(0.5-α, 0.5+α)	Indeterminate region
[0.5+α, 1]	Visible region

The only exception is when $\alpha = 0$, a probability of 0.5 is considered to be indeterminate to avoid confusion of whether it should belong to the Invisible or Visible region. Our classifier takes as input the extracted parameters, and applies Model 3. If the resulting probability of visibility does not fall in the indeterminate region, we decide the packet loss is visible or invisible appropriately.

To evaluate the accuracy of the model for classification purposes, we compute the ground truth regarding visibility using the results of the subjective test. Further, for the evaluation process, we only consider those packet losses where the ground truth regarding visibility is not indeterminate. Thus, we only consider those cases where both $\tilde{p}$ and $\hat{p}$ do not fall into the indeterminate region. A decision is correct if $\tilde{p}$ and $\hat{p}$ both fall into the visible region or the invisible region. A decision is wrong if $\tilde{p}$ falls in the invisible region and $\hat{p}$ falls in

α	Accuracy %		
	RR	NR-P	NR-B
0	0.845	0.848	0.849
0.05	0.862	0.859	0.856
0.10	0.89	0.884	0.882
0.15	0.901	0.906	0.907
0.20	0.946	0.96	0.959
0.25	0.96	0.972	0.969
0.30	0.975	0.98	0.978
0.35	0.989	0.994	0.988
0.40	0.995	0.995	0.991
0.45	0.996	0.993	0.993

TABLE IV
CROSS-VALIDATION ACCURACY WITH VARYING α

the visible region or vice-versa. Here, we assign zero cost to classifying an invisible/visible packet loss as an indeterminate packet loss, and unit cost for each wrong decision described above.

We vary α from 0 to 0.45 in steps of 0.05 and calculate the accuracy of the model for each value of α. Figures 5, 6 and 7 show the variation of cross-validation accuracy with α for the initial and final models, for the RR, NR-P, and NR-B methods, respectively. The final model is more accurate than the initial model in all the three cases.

Figure 8 compares the accuracy of the RR, NR-P and NR-B methods using the final model for different values of α, and Figure 9 shows the corresponding number of decisions in each case. The accuracy achieved by the final model for different values of α is also listed in Table IV. Clearly, all the three methods perform similarly. In particular, our NR-B method performs almost as well as our RR method. Further, fewer decisions are made as the size of indeterminate region (2α) increases, but their accuracy increases. If we choose a large value of α, we will obtain high accuracy but fewer decisions. On the other hand, a small value of α allows us to make more decisions, but with lower accuracy.

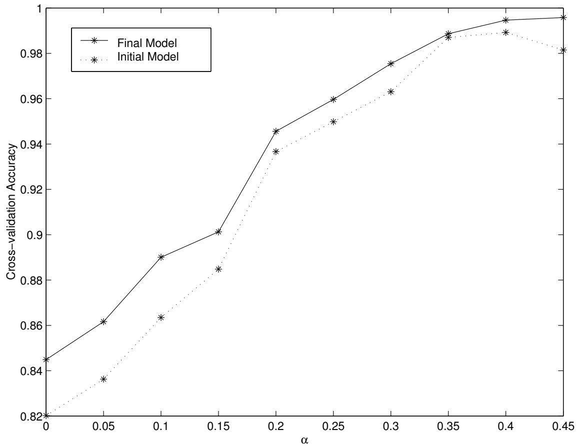

Fig. 5. RR: Cross-validation accuracy versus α

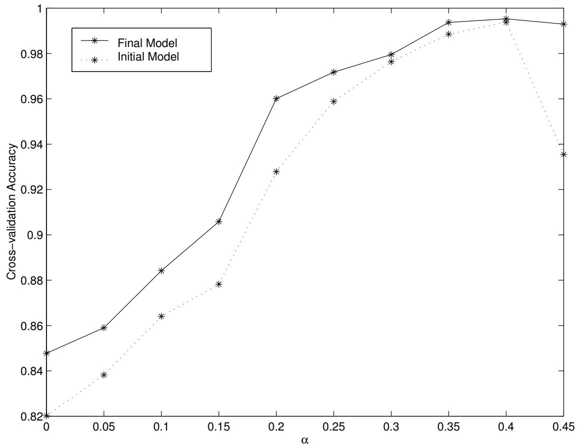

Fig. 6. NR-P: Cross-validation accuracy versus α

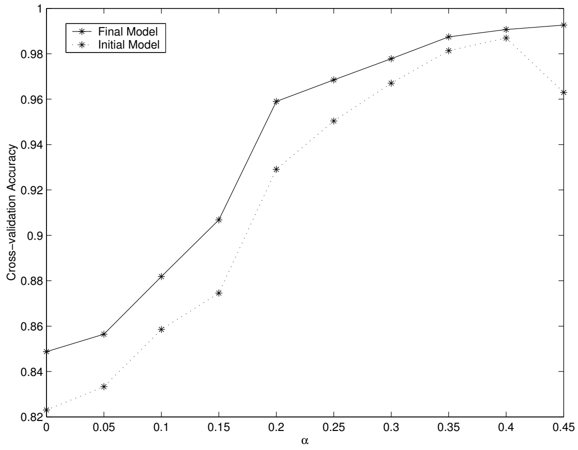

Fig. 7. NR-B: Cross-validation accuracy versus α

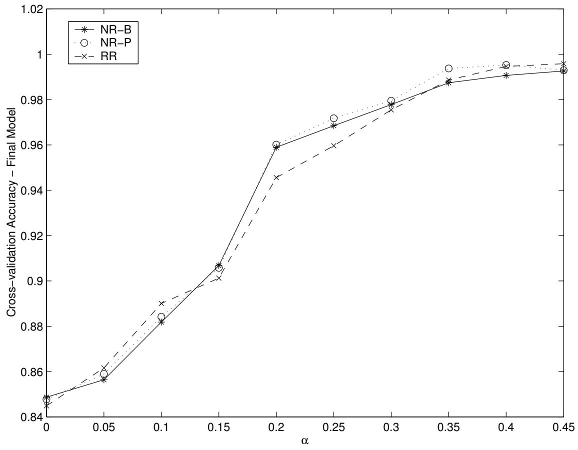

Fig. 8. Comparision of RR, NR-P and NR-B methods

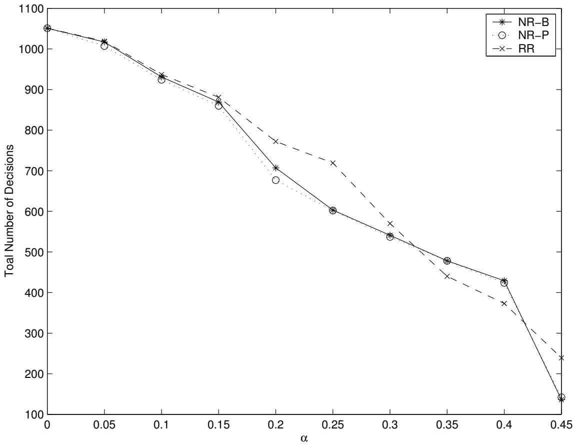

Fig. 9. Number of decisions versus α

VII. Conclusions

We considered the problem of predicting the probability that a packet loss is visible, using measurements from either the entire encoded video, the decoded video pixels, or just the received lossy bitstream. We used a logistic regression model to fit the data from subjective tests using these measurements. We examined how to describe pertinent factors such as motion to best predict visibility. As a result, we use MOTM instead of MOTX and MOTY and FRAMETYPE instead of TMDR, and we dropped insignificant factors such as MOTA. Finally, we used the predicted probabilities to decide whether a packet loss is visible or not. We achieved a high cross-validation accuracy of 96.9% in the NR-B case when $\alpha = 0.25$.

Our model may be useful in scenarios other than measuring video quality inside the network. For example, it may be useful in setting thresholds on allowable packet loss rate. Further, it could be used to prioritize packets within the network based on their probability of visibility, so as to achieve visually optimal streaming.

Acknowledgements

The authors would to like thank NSF and $CALIT^2$ for supporting this research through research grants. The authors would also like to thank Prof. Charles Berry at UCSD for his support and guidance with statistical methods.

References

[1] A. R. Reibman, S. Kanumuri, V. Vaishampayan and P. C. Cosman, “Visibility of Individual Packet Losses in MPEG-2 Video”, *IEEE ICIP*, October 2004 (accepted).

[2] P. McCullagh and J. A. Nelder, “Generalized Linear Models”, 2^{nd} Edition, Chapman & Hall.

[3] Verizon Laboratories (G. W. Cermak), “Videoconferencing Service Quality as a function of bandwidth, latency, and packet loss”, T1A1.3/2003-026, May 2003.

[4] B. Chen and J. Francis, “Multimedia Performance Evaluation”, AT&T Technical Memorandum, February 2003.

[5] S. Mohamed and G. Rubino, “A study of real-time packet video quality using random neural networks”, *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 12, no. 12, pp. 1071-1083, Dec 2002.

[6] 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 Processing*, vol. 2, no. 2, pp. 212-222, April 1993.

[7] A. Watson and M. A. Sasse, “Measuring perceived quality of speech and video in multimedia conferencing applications”, *ACM International Conference on Multimedia*, pp. 55–60, April 1998.

[8] M. Masry and S. Hemami, “Perceived quality metrics for low bit rate compressed video”, *Proc. IEEE ICIP*, vol. 3, pp. 49-52, June 2002.

[9] Z. Yu, H. R. Wu, S. Winkler and T. Chen, “Vision-model-based impairment metric to evaluate blocking artifacts in digital video”, *Proceedings of the IEEE*, vol. 90, no. 1, pp. 154-169, January 2002.

[10] S. Wolf and M. Pinson, “In-service performance metrics for MPEG-2 video systems”, IAB, Montreux, Switzerland, November 1998.

[11] A. R. Reibman, V. Vaishampayan and Y. Sermadevi, “Quality monitoring of video over a packet network”, *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 327-334, April 2004.

[12] M. S. Moore, J. Foley and S. Mitra, “Defect Visibility and content importance implications for the design of an objective video fidelity metric”, *Proc. IEEE ICIP*, vol. 3, pp. 45-48, June 2002.

[13] O. Verscheure, P. Frossard and M. Hamdi, “Joint Impact of MPEG-2 Encoding Rate and ATM Cell Losses on Video Quality”, *Global Telecommunications Conference (GLOBECOM)*, vol. 1, pp. 71-76, November 1998.

[14] P. Gastaldo, S. Rovetta and R. Zunino, “Objective Quality Assessment of MPEG-2 Video Streams by using CBP Neural Networks”, *IEEE Trans. on Neural Networks*, vol. 13, pp. 939-947, July 2002.

[15] A. E. Conway and Y. Zhu, “Applying Objective Perceptual Quality Assessment Methods in Network Performance Modeling”, *Proc. Eleventh Int’l Conf. on Computer Communications and Networks*, pp. 116-223, October 2002.

[16] The Website of R Project, http://www.r-project.org/