Network-Based H.264/AVC Whole-Frame Loss Visibility Model and Frame Dropping Methods

Yueh-Lun Chang, Ting-Lan Lin, *Member, IEEE*, and Pamela C. Cosman, *Fellow, IEEE*

***Abstract*— We examine the visual effect of whole-frame loss by different decoders. Whole-frame losses are introduced in H.264/AVC compressed videos which are then decoded by two different decoders with different common concealment effects: frame copy and frame interpolation. The videos are seen by human observers who respond to each glitch they spot. We found that about 39% of whole-frame losses of B frames are not observed by any of the subjects, and over 58% of the B frame losses are observed by 20% or fewer of the subjects. Using simple predictive features that can be calculated inside a network node with no access to the original video and no pixel level reconstruction of the frame, we develop models that can predict the visibility of whole B frame losses. The models are then used in a router to predict the visual impact of a frame loss and perform intelligent frame dropping to relieve network congestion. Dropping frames based on their visual scores proves superior to random dropping of B frames.**

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

I. Introduction

PACKET losses of compressed video during transmission in networks degrade the decoded video quality observed by the end users. Losses occur for different reasons. An intermediate router can drop packets because the incoming data rate is so high that the buffer overflows. With internet protocol television, a subscriber may want to watch a video in high resolution, but his access bandwidth may be less than required. In this situation, a router should drop a sufficient percentage of data to meet the access capabilities of the subscriber. The packet dropping rates required at the router can vary by a large amount. The packet dropping policy in the router should be intelligent enough to minimize the video quality damage observed by the end user.

Video quality monitoring in networks is an active research area. Some approaches predict the video quality using objective measures such as mean-squared error (MSE) or peak-to-signal noise ratio [1]–[4]. However, MSE is not well correlated with human perception [5]. Therefore, subjective tests collecting direct responses from individuals who watch impaired videos are necessary to understand how different packet losses are perceived by people. The work in [6] and [7] focused on modeling the average quality of videos as a function of average packet loss rate (PLR). In [8], the authors developed a model utilizing mismatched blocks to predict the subjective video quality. The scene complexity and level of motion are used to predict perceptual quality in [9].

These methods give an overall quality score for the sequence, but do not tell us how best to drop packets in the router to minimize video quality degradation during network congestion. In our previous work [10], packet dropping methods based on perceptual video quality were discussed. The visual importance of each packet is evaluated in the encoder by an *encoder-based* packet loss visibility model. All information available to the encoder can be used. Before the packet is sent to the network, a single bit of priority score is added to the header based on the estimated packet loss visibility. The router can then drop packets of lower priority during congestion. In [10], we showed that the dropping policy that uses visibility-based packet prioritization performs well compared to the common DropTail policy, and compared to a prioritization method based on the induced MSE if that packet is lost [11].

One limitation of [10] is that the priority score needs to be determined at the encoder and added as one bit to the packet header. In [12], we do not assume that packets coming into the router are embedded with a visual priority bit, for each packet, the visual importance is obtained by the *network-based model* described in [13], which only requires information extractable within one packet and no reference frame information. This is desired because, in a router, the incoming packets may be out of coding order or may be multiplexed with other video streams, so the router may not be able to identify which the reference packet of the current packet is. Also, we want the complexity of the factor extraction process to be low to be used in the network. Therefore, we do not consider factors such as initial MSE or scene-cut detection that require pixel domain reconstruction by full decoding as used in [10].

Also in [12], we devised a packet dropping method for widely varying PLRs including high rates. The packet loss visibility model in our prior work was designed for packets that contain individual slices (defined to be one horizontal row of macroblocks) of a frame. For these slice losses, after error concealment, spatial misalignment relative to the intact portion of the frame stands out. Spatial misalignment artifacts can be more distracting than temporal frame freeze [14]. Therefore

Manuscript received July 22, 2011; revised January 22, 2012; accepted February 29, 2012. Date of publication March 21, 2012; date of current version July 18, 2012. This work was supported in part by Intel Corporation, Cisco Systems, Inc., and the UC Discovery Grant Program. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Stefan Winkler.

Y.-L. Chang and P. C. Cosman are with the Department of Electrical and Computer Engineering, University of California, San Diego, CA 92093 USA (e-mail: yuc018@ucsd.edu; pcosman@ucsd.edu).

T.-L. Lin is with the Department of Electronic Engineering, Chung Yuan Christian University, Chungli 32023, Taiwan (e-mail: tinglan@cycu.edu.tw).

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.2012.2191567

TABLE I
SUMMARY OF THE SUBJECTIVE EXPERIMENT SETUP. H IS THE HEIGHT OF THE VIDEO

	SDTV
Resolution	720×480
Bitrate	2.1 Mb/s
H.264 profile	Main profile Level 3
Viewing distance	$6H$
Frame rate	30 ft/s
GOP	IBBPBBPBBPBBPBB 15/3

in [12], the algorithm drops the least visible *frames*, incurring fewer blocky artifacts compared to dropping on a *slice* basis. We showed that the frame-level temporal interpolation artifact is better than the slice-level spatial misalignment artifact using the video quality metric (VQM) [15]. VQM is a full-reference metric that considers jerky motion, blocking, and blurring [16], and has been shown to correlate well with human perception [17].

Nevertheless, the whole frame that is to be dropped in [12] was estimated by the network-based visibility model for single-slice packets described in [13]. That is, the visibility score for the frame was taken to be the sum of the visibility scores for the slices that compose the frame. And those visibility scores for slices came from a model designed using a human observer experiment on slice loss data, which do not directly reflect the frame loss visibility. This paper aims at obtaining and exploiting more meaningful scores for frame losses. We conduct a subjective experiment on whole-frame loss, and build a direct model for the loss. Two common concealment methods are used for whole-frame losses: frame copy and temporal frame interpolation. We analyze the experimental data, and model the whole-frame packet loss visibility based on information associated with the lost frames. We use the model to intelligently drop frames, and compare performance with [12] and [18].

Perceptual quality of frame losses is also discussed in the literature; [19] concludes that viewers preferred a single but long freeze event to frequent short freezes. In [20], different whole-frame loss types were studied as a function of frame-loss burst length and distribution. The authors conclude that the visibility of frame dropping is dependent on the content, loss duration, and motion. Later, in [21], they built an assessment model for subjective video quality as a function of frame-loss burst length and distribution. However, the quantities are computed in the pixel domain and require the original video, and the model aims to evaluate the quality of an entire lossy video but does not indicate the visual importance of a specific frame.

This paper is structured as follows. In Section II, the setup of the subjective experiment is introduced. Section III covers the analysis of data, and Section IV introduces the whole-frame loss modeling process and feature selection. Section V proposes frame dropping algorithms using the whole-frame loss visibility model and the frame size, and gives the performance of various methods. Section VI concludes this paper.

II. SUBJECTIVE EXPERIMENT ON WHOLE-FRAME LOSSES

In this section, we introduce the subjective experiment setup, including the encoding configuration, decoder concealment, and experimental design. The video encoder is H.264/AVC JM 9.3. Encoder settings (Table I) adhere to ITU and DSL forum recommendations [22], [23]. Each network abstraction layer (NAL) packet contains a horizontal row of macroblocks (16×16 pixels) in a frame. Our tested resolution is standard-definition television (720×480), so we have 30 packets per frame. Nine videos with widely varying motion and texture characteristics are concatenated into a 20-min sequence, and their descriptions are listed in Table II.

The decoders we considered are the JM 9.3 standard decoder [24] which produces frame copy artifacts, and FFMPEG [25] which conceals whole-frame losses using temporal frame interpolation. For the JM decoder, the lost frame is concealed by copying the pixels from the previous reference frame. For the FFMPEG decoder, a lost P frame is concealed by copying the pixels from the previous reference frame, and a lost B frame is concealed by temporal interpolation between the frame pixels of the previous and the future reference frames. These two decoders are widely used in academia and industry.

In this experiment, we concentrate on B frames. We introduce whole-frame loss events once every 4 s to allow observers enough time to respond to each individual loss event. There are two types of whole-frame loss events: single whole-frame loss and dual whole-frame loss, every loss event occurs in the first 3 s of each 4-s interval. Among these intervals, we uniformly inject single or dual whole-frame losses in a group of pictures (GOP) (in the dual cases, the *distance between the two lost frames* in one GOP could range from 1 to 13).

We create six different realizations of whole-frame loss events of the 20-min video, producing 900 distinct single whole-frame loss events and 900 dual whole-frame loss events. All the six lossy videos are decoded by FFMPEG and JM decoders. A subject watches two different loss realizations of whole-frame loss events from the same decoder, so a session involves 40 min of actual watching time per subject. The experiment takes 1 h or less, including an introductory session and a break. When viewers see a glitch, they respond to it by pressing the space bar. If the response is within 2 s of the loss, the loss event is regarded as visible. Each of the 40-min lossy videos is watched by 10 people.

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

III. DATA ANALYSIS

In this section, we analyze the two types of whole-frame loss events: single frame losses and dual frame losses.

TABLE II
DESCRIPTION OF VIDEO CLIPS USED FOR THE SUBJECTIVE TEST

1	Earth	Nature documentary of wildlife in slow motion
2	Indianapolis	Crowds moving in an arena with some car racing scenes
3	Formula	Racing cars on a racetrack
4	New York	Introduction to a city with bird-eye and street views
5	Air show	Aair show scene with planes flying over the sky, and some audience on the ground
6	Golf	Broadcast golf game
7	Hawaiian	Hawaiian tourism of various scenes in shops and streets with panning camera
8	Soccer	High-motion beach soccer game with crowded people in the background
9	Stories	Daily life such as friends talking and family reunion

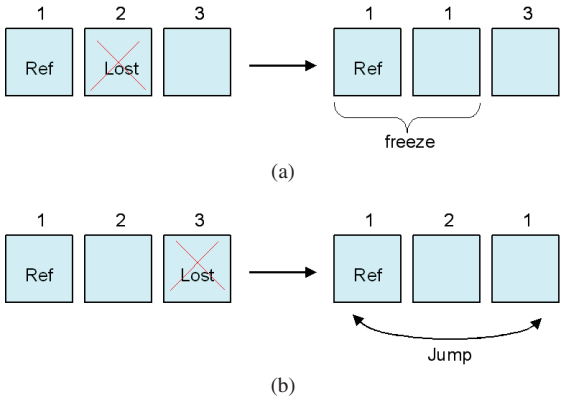


Fig. 1. Different visual effects by frame copy concealment. (a) Freeze effect. (b) Jump effect.

We examine the artifacts caused by the different concealment methods of the JM and FFMPEG decoders, and then compare the performance of the decoders.

A. Concealment Methods of the Decoders

JM uses frame copy and FFMPEG uses temporal interpolation for whole-frame loss concealment. For all B frames, JM conceals them by copying the previous intact reference frame, causing two types of temporal concealment artifacts: freeze and jump. For example, in Fig. 1(a), Frame 2, if lost, is concealed by copying Frame 1, the visual artifact is a short freeze because Frame 1 is displayed twice, in two consecutive frame time slots. In contrast, in Fig. 1(b), if Frame 3 is lost, it is also concealed by copying Frame 1. The displayed frames are 1 then 2 then 1, which causes jerkiness or jumping visually. The FFMPEG decoder conceals B frames by temporal interpolation most of the time, except for B frames after an IDR frame which are concealed by copying the IDR frame. For temporal interpolation, ghosting artifacts may appear when there is enough motion. The above three types of artifacts are called *"freeze," "jump,"* and *"interpolation"* effects. A visual example is demonstrated in Fig. 2. Frame 35 of the video sequence Stefan is lost and concealed by JM with frame copy in Fig. 2(a) and by FFMPEG with temporal frame interpolation in Fig. 2(b).

Table III shows the mean visibility for the three types of effects, calculated from the single whole-frame loss events.

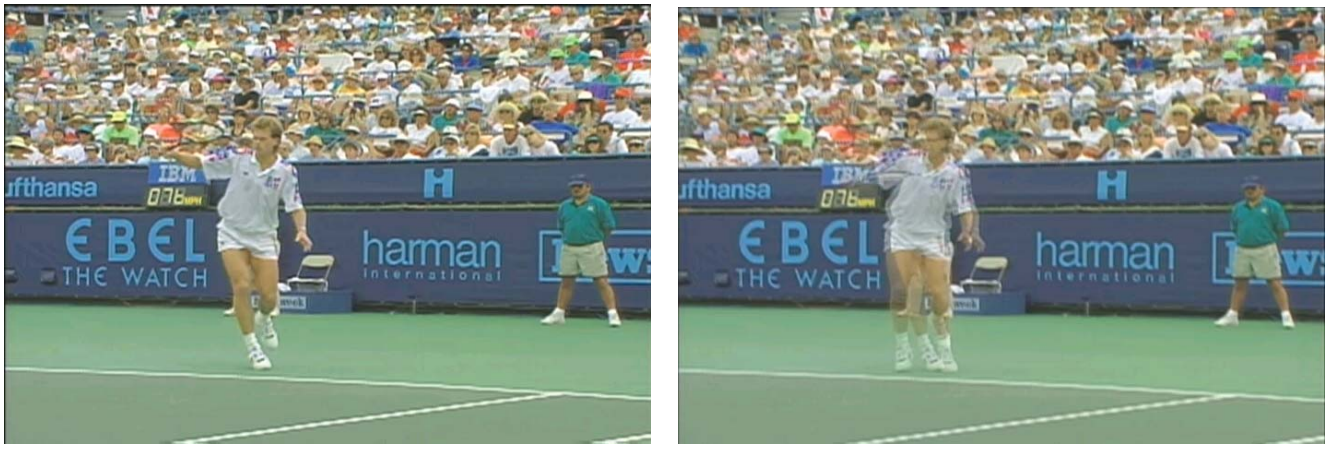
TABLE III
THREE TYPES OF ARTIFICIAL EFFECTS AND THEIR CORRESPONDING MEAN VISIBILITY, CALCULATED FROM THE SINGLE WHOLE-FRAME LOSS EVENTS

Effects	Mean visibility
Freeze	0.07
Jump	0.29
Interpolation	0.19

The freeze effect has the lowest mean visibility of 0.07, the jump effect has the highest of 0.29, and the visibility of interpolation is intermediate at 0.19.

Table IV summarizes all the possible artifacts of dual whole-frame loss concealment for each decoder, as well as the corresponding mean visibility for each. Fig. 3 shows the visibility versus different concealment artifacts. What is plotted in each case is the mean visibility together with the 95% confidence interval. The cross markers are for single frame losses, the circle markers are for JM dual frame losses, and the triangle markers are for FFMPEG dual frame losses. The 95% confidence intervals for the single frame loss concealments are nonoverlapping, meaning that the three effects (freeze, jump, and interpolation) have significantly different visibility. On the other hand, some of the 95% confidence intervals for the dual frame loss concealments are overlapping because the artifacts are the combination of two effects. The artifacts with jump effect have relatively high mean visibility, while the artifact with mere freeze effect has the lowest visibility. The loss events with interpolation effect give an intermediate result. About 30% of events are not seen by any observers, and on average 2.4 out of 10 observers see a dual frame loss event.

We also look at the dual frame loss visibility versus frame distance, as plotted in Fig. 4. In our experiment, the frame distance for the two nearby whole-frame losses in one GOP ranges from 1 to 13. The mean visibility is periodically higher for frame distance equal to 4, 7, 10, and 13. In the dual frame loss events, for a certain frame distance there are several possible frame loss combinations, which result in different artifacts since the concealments are not the same. For instance, when frame distance equals 1, the visual artifacts are either two freeze effects or two interpolation effects. The mean visibility of each frame distance is a weighted average of the visibility for the various dual frame loss concealments which can occur



(a) (b)

Fig. 2. Frame 35 of the video sequence "Stefan" is lost and concealed by the (a) JM decoder with frame copy and (b) FFMPEG decoder with temporal frame interpolation.

TABLE IV
POSSIBLE ARTIFACTS FOR CONCEALED DUAL WHOLE-FRAME LOSSES AND THE CORRESPONDING MEAN VISIBILITY FOR BOTH JM AND FFMPEG DECODERS

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

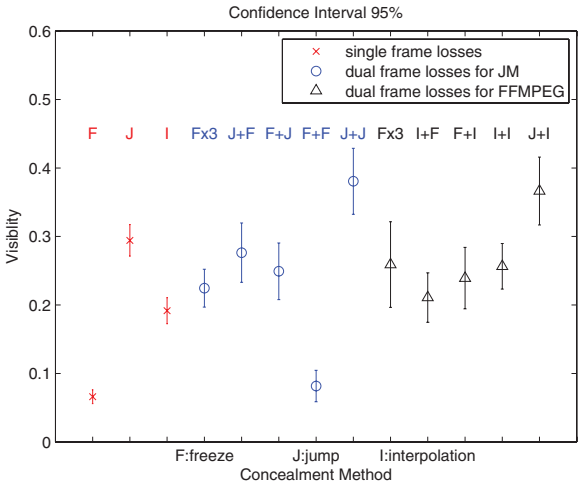


Fig. 3. Whole-frame loss visibility showing the mean and 95% confidence intervals for different concealment artifacts.

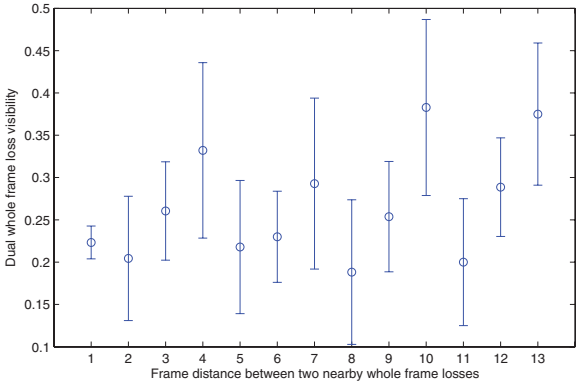


Fig. 4. Dual whole-frame loss visibility showing means and 95% confidence intervals, for every frame distance.

at that spacing. Statistically, when frame distance equals 4, 7, 10, and 13, their frame loss combinations result in a larger percentage of jump effect compared to other frame distance cases, and it makes these four frame distance cases to have higher mean visibility.

Another way to analyze the visibility is to group events into adjacent dual frame losses and separate dual frame losses. The two lost frames are adjacent if the frame distance equals 1, while they are separate if the frame distance is greater than 1. Fig. 5 shows the dual frame visibility for the adjacent and separate cases. It is apparent that adjacent dual frame losses have lower visibility than separate dual frame losses since the adjacent cases lead only to the two less visible effects (freeze and interpolation) while the separate cases can lead to the jump effect.

B. Comparison of the Decoders

In this section, we compare the performance of the JM and FFMPEG decoders only for single frame losses since we would like to build models that predict visibility for an isolated frame loss. Fig. 6 shows the histograms of the single whole-frame loss visibility of the JM and FFMPEG decoders. For the JM decoder, 40.8% of the losses are not observed by any

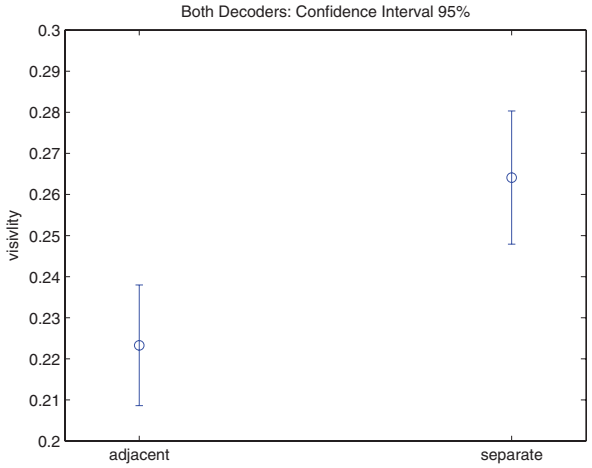


Fig. 5. Dual whole-frame loss visibility showing means and 95% confidence intervals for the adjacent and separate cases.

subjects (visibility is zero), and 62.4% of losses are seen by 2 or fewer out of 10 people (i.e., have visibility less than or equal to 0.2). For the FFMPEG decoder, 38.9% of the losses are not observed by any subjects, and 58.3% have visibility less than or equal to 0.2. One implication is that, if we can identify the frames that are less visible to viewers when lost, in the case of network congestion, we can choose to drop unimportant frames to relieve network congestion and not many end users will observe the losses.

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

Fig. 7 is the 3-D histogram of the single whole-frame loss visibility with respect to the JM and FFMPEG decoders. This figure shows that the invisible whole-frame losses decoded by JM usually are also invisible by FFMPEG and vice versa. The JM decoder has a better score than FFMPEG on 33.2% of cases, and FFMPEG has a better score 29.6% of the time. The remaining 37.2% of the whole-frame losses are observed by exactly the same number of observers for JM and FFMPEG. Among the tie cases, 79% represent losses with zero visibility for both decoders. The average whole-frame loss visibility over all the data is 0.1716 for JM and 0.1879 for FFMPEG, indicating that, on average, whole-frame losses concealed by JM are slightly less visible than by FFMPEG.

For a significance test between the visibility scores of FFMPEG and JM, we cannot perform a hypothesis test that assumes the data to be normal (e.g., t-test) since from Fig. 6 their distribution is far from normal. Therefore, we resort to nonparametric hypothesis testing. The Wilcoxon signed rank test (paired comparison) [26] compares paired data x and y in a two-sided test, where the null hypothesis H_0 is that the median of $x - y$ is zero, against the alternative that the distribution does not have zero median. Let x_i and y_i be the visibility for FFMPEG and JM in the ith comparison set. Define $w = \sum_{i=1}^{n} r_i z_i$, where r_i is the rank of $|x_i - y_i|$ among all $|x_j - y_j|$, and $z_i = 1$ if $x_i - y_i > 0$ and $z_i = 0$ otherwise. Here, $n = 900$, which is the number of losses. The statistic for the test

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

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

IV. Whole-Frame Packet Loss Visibility Model

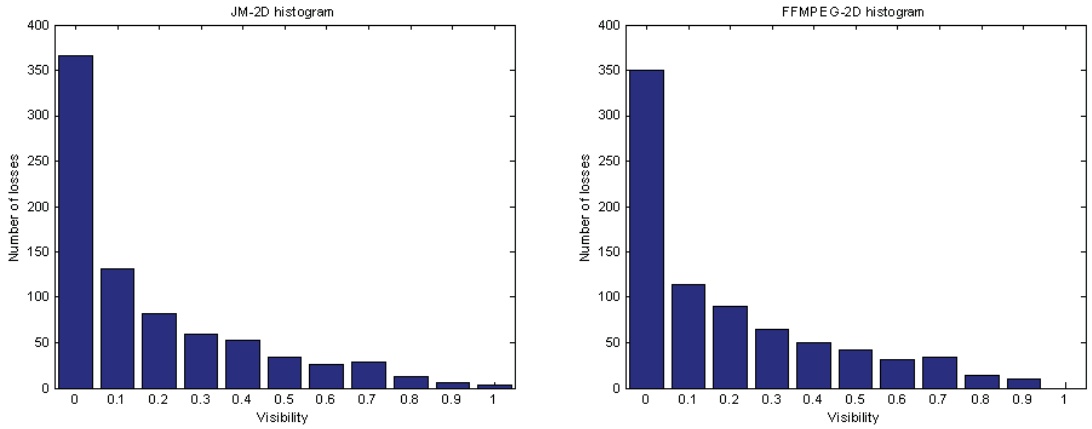
In this section, we construct a prediction model for whole-frame loss visibility using the data from the single whole-frame loss events. To predict the loss visibility, we consider network-extractable factors associated with a particular frame computed from a bit stream. The process of model building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

A. Factors Extractable from Bit Stream for Predicting Frame Loss Visibility

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

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

In addition, MB modes might affect the frame loss visibility, therefore we include the number of MBs that are coded as INTRA (**NumIntraMB**), INTER (**NumInterMB**), DIRECT (**NumDirectMB**) and SKIP (**NumSkipMB**) as model factors.



(a) (b)

Fig. 6. Histogram of single whole-frame loss visibility. (a) By JM Decoder. (b) By FFMPEG decoder.

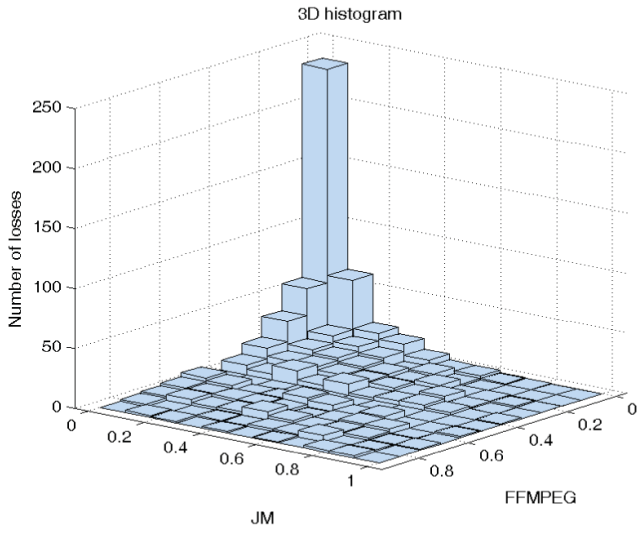


Fig. 7. 3-D histogram of single whole-frame loss visibility by JM and FFMPEG decoders.

To include in a simple way the effects of concealment, we defined Boolean factors **IsFreezeByJM**, **IsJumpByJM**, **IsInterpolation**, **IsFreezeByFFMPEG**, and **IsJumpByFFMPEG**, which are set when a certain effect is possibly present for a frame. These five concealment-related factors could be obtained by knowing the temporal location of a frame.

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

B. Modeling Process

In the experiment and data analysis, we assume that each viewer's response is an independent observation of the average viewer (for whom we are developing the model). Therefore, each viewer response can be considered i.i.d. with probability p for seeing a particular packet loss. Generalized linear models (GLMs) are an extension of classical linear models [27], [28]. The probability of visibility is modeled using logistic regression, which is a type of GLM that is a natural model to predict the parameter p of a binomial distribution [27]. 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 GLM can be represented as

$$g(p_i) = \alpha + \sum_{j=1}^{P} x_{ij}\beta_j \tag{2}$$

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 α 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

$$g(p) = \log\left(\frac{p}{1-p}\right). \tag{3}$$

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

TABLE V
TABLE OF FACTORS IN THE ORDER OF IMPORTANCE FOR AVG_JM_FFMPEG MODEL

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

TABLE VI
TABLE OF FACTORS IN THE ORDER OF IMPORTANCE FOR MAX_JM_FFMPEG MODEL

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

an iterative feature selection technique is implemented by MATLAB.

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

If we have information about the user and know the exact decoder to be deployed, we could build models based on different decoders: JM_Model and FFMPEG_Model. Fig. 8(a) and (b) shows the plots of deviance versus the number of factors included in the model. The concealment-related factors greatly improve the deviance. Because most of the losses in the FFMPEG_Model are concealed by temporal interpolation with interpolation effect, the concealment-related factors benefit the JM_Model more because they correctly depict the visual effects of freeze or jump, which are both caused by frame copy but with very different influence on the visibility.

In case one does not know at an intermediate router which decoder will be used ultimately at the receiver side, it is desirable to develop one model to predict the whole-frame packet loss visibility for both decoders. The data is combined in two ways: taking the average of the JM and FFMPEG visibility scores associated with the same whole-frame loss, and taking the maximum of the JM and FFMPEG visibility scores. The latter aims to predict the worst case visibility. The factors in order of importance and the corresponding coefficients of the final models of Avg_JM_FFMPEG and Max_JM_FFMPEG are listed in Tables V and VI, respectively. Their plots of deviance versus the number of factors included are shown in Fig. 8(c) and (d).

The first seven important factors are almost the same for both models, but with a slightly different order. More than 70% of factors in the model involve motion vector computations. This indicates the amount of motion in the lost frame dominates the visual performance. Fig. 9 shows the scatter plots of visibility score versus three of the top important factors MeanMotM, VarMotX, and VarMotY. Since the visibility scores take on only 11 discrete values (0, 0.1, 0.2, . . . 1) which cause the dots to overlap in the scatter plot, we add random values between 0 and 0.095 to each visibility score for plotting. So the points with visibility score of 0 are shown with y values randomly between 0 and 0.095, those with values of 0.1 are shown with y values in the range of 0.1–0.195, etc. This makes it easier to see the distinct dots. The trend in the plots shows that the visibility tends to be larger when the three factors have higher value, the dots tend to be more tightly clustered at the low-visibility side when the factor values are small. As in the separate model, the concealment-related factors are important. Without these concealment-related factors, the best deviance for the Avg_JM_FFMPEG and Max_JM_FFMPEG models are only 171 and 229, which are considerably higher than when concealment-related factors are included. The nine video clips used in the subjective experiment included both high and low motion, we found that the model accuracy was slightly higher for slow-motion clips than for high-motion clips.

V. WHOLE-FRAME DROPPING

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

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

A. Dropping Algorithms Under Comparison

We use the proposed models in Tables V and VI that directly predict the whole-frame visibility to perform the frame

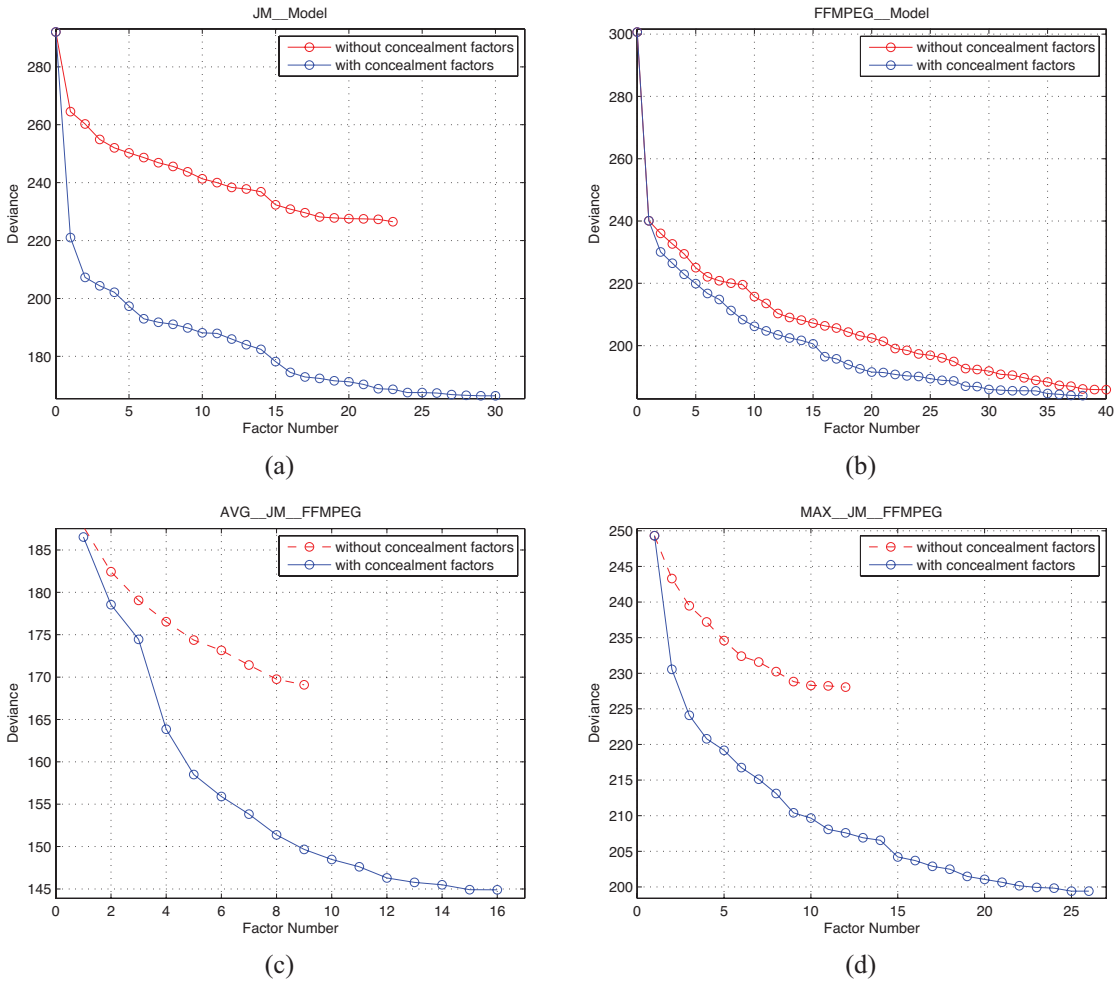


Fig. 8. Deviance reduction as additional factors are included in the (a) JM_Model, (b) FFMPEG_Model, (c) Avg_JM_FFMPEG, and (d) Max_JM_FFMPEG model.

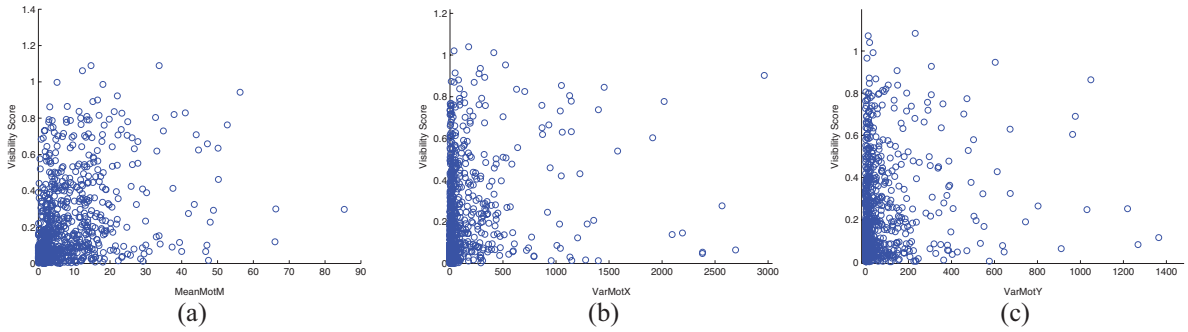


Fig. 9. Scatter plots of visibility score versus three of the top important factors. (a) MeanMotM. (b) VarMotX. (c) VarMotY.

importance estimation. The model in Table V is used to predict the frame importance and drop frames until the target BRR is achieved. This method is denoted **FrameMean**. When the model in Table VI is used, we denote it **FrameMax**.

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

As a baseline for comparison, [18] discusses a dropping method that is implemented in a video-aware digital subscriber line access multiplexer. It inspects the nal_ref_idc bit in every NAL unit header. Packets that do not serve as reference pictures can be dropped during network congestion. That corresponds to B frames in our case. We simulate this method by randomly dropping B frames until the BRR is achieved. We denote this method by **RandomBFrame** and

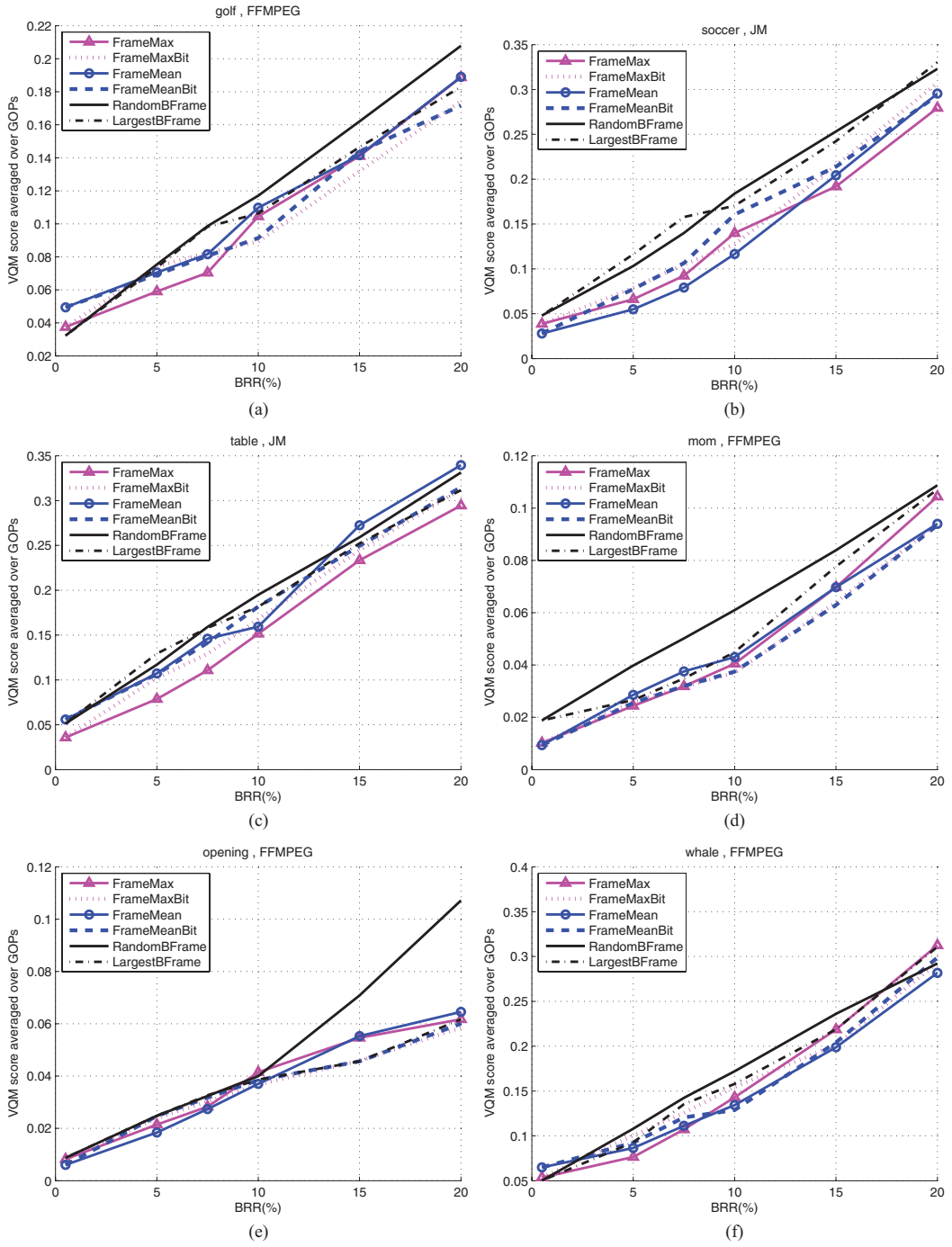


Fig. 10. Average VQM score over GOPs versus BRR for the six packet dropping policies for (a) FFMPEG for *Golf*, (b) JM for *Soccer*, (c) JM for *Table tennis*, (d) FFMPEG for *Mother Daughter*, (e) FFMPEG for *Opening*, and (f) FFMPEG for *Whale*. Lower VQM scores correspond to higher quality.

define its performance by the results averaged over 50 random realizations. A variation that considers size drops B frames in descending order of size. This dropping method is denoted **LargestBFrame**.

B. Experimental Results

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

The video encoder is H.264/AVC JM9.3. The resolution is SDTV. The tested videos are encoded at 2.5 Mb/s, 30 ft/s using main profile level 3. The GOP structure is IBBP (18 frames). We perform each dropping algorithm in a GOP, and the BRR is the percentage of bits to be dropped for this GOP. After the dropping policy is performed for a GOP, the FFMPEG and JM decoding, and their corresponding error concealment are run, and then the VQM [15] is calculated to obtain the video quality score for this lossy GOP. VQM is a full-reference metric that ranges from 0 (excellent quality) to 1 (poorest possible quality).

Eight videos are tested in the simulation, they contain a wide variety of scenes with different types of camera motion, object motion, and spatial texture. *Golf* has slow movement, and *Soccer* has fast motion, these two videos are among the sequences used in the subjective experiment in Section II. Other clips are *News*, *Mother Daughter*, and *Opening*, which have low motion, and *Stefan*, *Table Tennis*, and *Whale* with high motion, these standard test videos were not used in the subjective experiments.

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

Fig. 10(a) shows the VQM score averaged over GOPs versus BRR for the six dropping methods for the video *Golf*, where the lossy bitstream is decoded by FFMPEG. We see that, as the BRR goes up, the video quality deteriorates (the VQM scores go up).

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

We then compare the visibility-based methods. FrameMean and FrameMax perform better than our previous method in [12], which means the models directly built from whole-frame losses provide a better prediction of frame importance than estimating frame importance by summing the visibility of slices in a frame using the slice loss visibility model. In addition, the visibility-per-bit methods provide further improvement. FrameMeanBit and FrameMaxBit are better than FrameMean and FrameMax, respectively. These trends can be observed in Fig. 10(a).

For all other videos shown from Fig. 10(b)–(f), there are similar trends. For the comparison between the visibility and the visibility-per-bit methods, it is not consistent that one of them is superior; however, in more than half of the cases, the visibility-per-bit method outperforms the visibility method.

Comparing the low-motion clips (*Golf*, *News*, *Mother Daughter*, and *Opening*) and the high motion ones (*Soccer*, *Stefan*, *Table Tennis*, and *Whale*), the slow-movement clips have lower VQM scores than the fast-movement clips for a given BRR. In the simulation, the highest VQM scores for the fast clips are more than 0.3, while the highest scores for the slow ones are less than 0.25. This indicates that the losses are more concealable for *Golf*, *News*, *Mother Daughter*, and *Opening*. Comparing the best dropping approach with the worst one, the fast-motion videos have larger gains. In Fig. 10(b), (c), and (f), the improvement for high-motion videos increases more, up to $0.05-0.08$ VQM score, whereas the slower videos have less than 0.04 VQM score gain as in Fig. 10(a), (d), and (e).

VI. Conclusion

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

1) When isolated B frames were lost and concealed by either the JM standard decoder or the FFMPEG decoder, about 40% of such losses were not seen by any of the 10 observers, and about 60% of such losses were seen by 2 or fewer out of 10 observers. This suggests that whole-frame loss of isolated B frames is highly concealable.
2) Although the JM and FFMPEG decoders had very similar overall performance, this result hides the fact that, depending on frame position, JM concealment produces freeze or jump artifacts, whereas FFMPEG concealment produces mostly interpolation artifacts (and only rarely a freeze or jump artifact if a B frame after an IDR is lost). These concealment approaches do not have similar performance, as freeze is the least noticeable and jump is the most visible.
3) When two B frames are lost within the same GOP, about 30% of such events are not seen by any observers. On average, 2.4 out of 10 observers see a dual frame loss event. The least visible type of dual frame loss event consists of two isolated freeze artifacts. So if a router needs to drop two frames within a GOP, the best choice would be to have two separate pairs of B frames in a GOP each suffering the loss of the first B frame in the pair. This leads to the least visible type of loss for the JM decoder, and among the least visible for FFMPEG.
4) Visibility models that are specific for the JM and FFMPEG decoders are more successful at predicting the frame loss visibility than are models which attempt to predict the average or the worst case of the two decoders. Nonetheless, a model designed to predict the average visibility score can provide improved frame dropping decisions compared to random B frame dropping, and compared to slice-based visibility dropping decisions from [12].
5) In the condition, where an intermediate router is congested and is forced to drop frames (needing to achieve some target BRR), if for complexity reasons one does not wish to drop frames using the visibility model, there are still ways to improve over random B frame dropping. One way is to drop the largest B frames until the target is met, this offers improvement especially for larger BRRs because one achieves the target with a smaller number of total frames dropped. A second simple way to improve over random B frame dropping is to avoid dropping the second B frame in any pair of two consecutive B frames (this avoids the jump concealment artifact).

ACKNOWLEDGMENT

Preliminary results of this paper were presented in [29], Sections II and III-B were partly taken from [29], and Sections III-A, IV, and V represent new results.

REFERENCES

[1] 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, Apr. 2004.

[2] S. Tao, J. Apostolopoulos, and R. Guerin, "Real-time monitoring of video quality in IP networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 5, pp. 1052–1065, Oct. 2008.

[3] M. Naccari, M. Tagliasacchi, and S. Tubaro, "No-reference video quality monitoring for H.264/AVC coded video," *IEEE Trans. Multimedia*, vol. 11, no. 5, pp. 932–946, Aug. 2009.

[4] A. Eden, "No-reference estimation of the coding PSNR for H.264-coded sequences," *IEEE Consumer Electron. Trans.*, vol. 53, no. 2, pp. 667–674, May 2007.

[5] B. Girod, *What's Wrong with Mean-Squared Error*. Cambridge, MA: MIT Press, 1993.

[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 Process.*, vol. 2, no. 2, pp. 212–222, Apr. 1993.

[7] K. Yamagishi and T. Hayashi, "Parametric packet-layer model for monitoring video quality of IPTV services," in *Proc. IEEE Int. Conf. Commun.*, May 2008, pp. 110–114.

[8] N. Montard and P. Bretillon, "Objective quality monitoring issues in digital broadcasting networks," *IEEE Trans. Broadcast.*, vol. 51, no. 3, pp. 269–275, Sep. 2005.

[9] J. Hu and H. Wildfeuer, "Use of content complexity factors in video over ip quality monitoring," in *Proc. Int. Workshop Quality Multimedia Exp.*, San Diego, CA, Jul. 2009, pp. 216–221.

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

[11] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 207–218, Apr. 2006.

[12] T.-L. Lin and P. Cosman, "Packet dropping for widely varying bit reduction rates using a network-based packet loss visibility model," in *Proc. Data Comp. Conf.*, Snowbird, UT, Mar. 2010, pp. 445–454.

[13] T.-L. Lin and P. Cosman, "Network-based packet loss visibility model for SDTV and HDTV for H.264 videos," in *Proc. IEEE Acoust. Speech Signal Process. Int. Conf.*, Dallas, TX, Mar. 2010, pp. 906–909.

[14] N. Staelens, B. Vermeulen, S. Moens, J.-F. Macq, P. Lambert, R. Van de Walle, and P. Demeester, "Assessing the influence of packet loss and frame freezes on the perceptual quality of full length movies," in *Proc. Int. Workshop Video Process. Quality Metrics Consumer Electron.*, 2009.

[15] *The Website for VQM Software* [Online]. Available: http://www.its.bldrdoc.gov/n3/video/vqmsoftware.htm

[16] H. Himmanen, M. M. Hannuksela, T. Kurki, and J. Isoaho, "Objectives for new error criteria for mobile broadcasting of streaming audiovisual services," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 13, pp. 1–161, 2008.

[17] M. H. Loke, E. P. Ong, W. Lin, Z. Lu, and S. Yao, "Comparison of video quality metrics on multimedia videos," in *Proc. IEEE Image Process. Int. Conf.*, Oct. 2006, pp. 457–460.

[18] *Alcatel-Lucent Technical Paper: Access Network Enhancements for the Delivery of Video Services*, Alcatel-Lucent, Paris, France, May 2005.

[19] Q. Huynh-Thu and M. Ghanbari, "Impact of jitter and jerkiness on perceived video quality," in *Proc. Int. Workshop Video Process. Consumer Electron.*, 2006, pp. 1–6.

[20] R. Pastrana-Vidal, J. Gicquel, C. Colomes, and H. Cherifi, "Sporadic frame dropping impact on quality perception," *Proc. SPIE*, vol. 5292, pp. 182–193, Jan. 2004.

[21] R. Pastrana-Vidal and J. Gicquel, "Automatic quality assessment of video fluidity impairments using a no-reference metric," in *Proc. Int. Workshop Video Quality Metrics*, Jan. 2006, pp. 1–6.

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

[23] T. Rahrer, R. Fiandra, and S. Wright, "Triple-play services quality of experience (QoE) requirements," Architecture Transport Working Group, DSL Forum, Tech. Rep. TR-126, Dec. 2006.

[24] *H.264/AVC JM Software*, [Online]. Available: http://iphome.hhi.de/suehring/tml/

[25] *The Official Website of FFMPEG* [Online]. Available: http://ffmpeg.org/

[26] R. Larsen and M. Marx, *An Introduction to Mathematical Statistics and Its Applications*, 4th ed. Englewood Cliffs, NJ: Prentice-Hall, 2010.

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

[28] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, 2nd ed. New York: Wiley, 2000.

[29] T.-L. Lin, Y.-L. Chang, and P. C. Cosman, "Subjective experiment and modeling of whole frame packet loss visibility for H.264," in *Proc. IEEE Packet Video Conf.*, Dec. 2010, pp. 186–192.

Yueh-Lun Chang received the B.S. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2006, and the M.S. degree in electrical and computer engineering from the University of California, San Diego, in 2010, where she is currently pursuing the Ph.D. degree in electrical and computer engineering.

She interned with the Video System Group and Mixed-Signal Team, Qualcomm, San Diego, in 2009 and 2010, respectively. Her current research interests include video streaming in lossy networks, video error concealment, and perceptual video quality.

Ting-Lan Lin (S'08–M'11) received the B.S. and M.S. degrees in electronic engineering from Chung Yuan Christian University, Chung Li, Taiwan, in 2001 and 2003, respectively, and the Ph.D. degree in electrical and computer engineering from the University of California, San Diego, in 2010.

He interned with the Display System Group, Qualcomm, San Diego, in 2008. He has been an Assistant Professor with the Department of Electronic Engineering, Chung Yuan Christian University, since February 2011. His current research interests include video compression, video streaming in lossy networks, optimization of packet prioritization, and perceptual video quality.

Pamela C. Cosman (S'88–M'93–SM'00–F'08) received the B.S. degree (hons.) 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 Post-Doctoral Fellow with Stanford University and a Visiting Professor with the University of Minnesota, Minneapolis, from 1993 to 1995. In 1995, she joined the Faculty of the Department of Electrical and Computer Engineering, University of California at San Diego, where she is currently a Professor and Vice-Chair. She was the Director of the Center for Wireless Communications, University of California, from 2006 to 2008. Her current research interests include image and video compression and processing and wireless communication.

Dr. Cosman was a recipient of the ECE Departmental Graduate Teaching Award, a Career Award from the National Science Foundation, a Powell Faculty Fellowship, and a Globecom Best Paper Award in 2008. 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 Information Theory Workshop, San Diego, in 1998. She was an Associate Editor of the IEEE COMMUNICATIONS LETTERS and the IEEE SIGNAL PROCESSING LETTERS from 1998 to 2001 and from 2001 to 2005, respectively. She was the Editor-in-Chief as well as a Senior Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS from 2003 to 2005, from 2006 to 2009, and from 2010 to the present. She is a member of Tau Beta Pi and Sigma Xi.