SUBJECTIVE EXPERIMENT AND MODELING OF WHOLE FRAME PACKET LOSS VISIBILITY FOR H.264

Ting-Lan Lin, Yueh-Lun Chang and Pamela C. Cosman

ECE, University of California, San Diego

ABSTRACT

Whole frame losses are introduced in H.264 compressed videos which are then decoded by two different decoders with different common concealment effects. The videos are seen by human observers who respond to each glitch they spot. We found that about 38% of whole frame losses of B frames are not observed by any of the subjects, and well over 58% of the B frame losses are observed by 20% or fewer of the subjects. Using simple predictive features which can be calculated inside a network node with no access to the original video and no pixel level reconstruction of the frame, we developed a model which can predict the visibility of whole frame losses in B frames. This model could be useful for designing an intelligent frame dropping approach for use at a router during congestion.

Index Terms— H.264, packet loss, frame interpolation, frame freeze, error concealment, subjective metric.

1. INTRODUCTION

Compressed video transmission in networks suffers from packet losses, which affect the decoded video quality to different degrees. It is important to characterize the packet loss impact in terms of video quality. Traditionally, objective measures like MSE (mean-square error) or PSNR are used as indicators of video quality. However, MSE is not correlated with human perception well. Therefore subjective tests collecting direct responses from subjects who watch impaired videos is necessary to understand how different packet losses are perceived. In our prior studies [1, 2], we reported subjective test results and built packet losses visibility models to predict the visual importance for each packet when it is lost and concealed by the decoder. In [1, 3], we show that in the case of network congestion, the router can choose the best packets to drop according to our model and achieve a better visual quality, compared to existing methods in the literature [4] and in industry [5].

However, the packet loss visibility modeling in our prior work was designed for packets that are just slices (defined to be one horizontal row of macroblocks) of a frame. For these types of packet losses, after error concealment, spatial misalignment relative to the intact portion of the frame stands out. Spatial misalignment artifacts can be more distracting than temporal frame freeze [6]. In [3], under the same dropping size constraint, we dropped packets on a slice basis, and on a frame 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) [7]. VQM is a full-reference metric developed by the National Telecommunication and Information Administration that has been shown to be better correlated with human perception than other full reference video quality metrics [8].

Nevertheless, which whole frame to be dropped in [3] was estimated by the visibility model for single-slice packets. That is, the visibility score for the frame was taken to be simply the sum of the visibility scores for the slices which compose the frame. And those visibility scores for slices came from a model designed using a human observer experiment involving slice loss data. Therefore, to obtain more meaningful scores for frame losses, in this paper, we conduct a subjective experiment to concentrate on the subjective results for whole frame loss, and build a direct model for whole frame loss. Two common concealment methods for whole frame losses are frame freeze and temporal frame interpolation. In this experiment, we simulate frame freeze by the frame copy error concealment in the JM standard decoder [9], and frame interpolation by FFMPEG [10]; these two decoders are popular in research and industry. In this paper we analyze the experimental data, and model the whole frame packet loss visibility based on information associated with the lost frames.

We hope to build a model that is suitable for router operation so that in the case of network congestion, the router is able to decide based on our model which frame or frames to drop to relieve the congestion while maintaining good video quality. Therefore, as in [3], we consider factors associated with the frame considered for dropping to be self-contained, meaning that the computation of the factors does not need other (reference) packets. This is desired since in a router, the incoming packets may be out of coding order or may be multiplexed with other video streams, so the router may not be able to identify which is the reference packet of the current packet. Also we want the complexity of the factor extraction process to be low. Therefore we do not consider factors such

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

as initial mean square error or scene cut detection that require pixel domain reconstruction by full decoding as used in [1].

Perceptual quality of frame losses is discussed in [11]. The work studies different whole frame loss type as a function of frame loss burst length and frame loss burst distribution. The authors conclude that the visibility of frame dropping is dependent on content, loss duration and motion. Later, in [12], they built an assessment model for subjective video quality as a function of frame loss burst and frame loss burst distribution. However, the quantities are computed in the pixel domain and require the original video. And the model aims to evaluate the quality of a lossy video, and does not indicate the visual importance of a specific frame.

This paper is structured as follows: in Section 2, the setup of the subjective experiment is introduced. Section 3 covers the analysis of data, and Section 4 introduces the whole frame loss modeling process and feature selection. Section 5 concludes the paper.

2. 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 JM 9.3. Encoder settings (Table 1) adhere to ITU and DSL Forum Recommendations [13, 14]. Each Network Abstraction Layer (NAL) packet contains a horizontal row of Macroblocks (16×16 pixels) in a frame. Our tested resolution is SDTV (720×480), so we have 30 packets per frame. Nine videos with widely varying motion and texture characteristics are concatenated into a 20-minute sequence.

The decoders we considered are the JM 9.3 standard decoder [9] which produces frame freeze artifacts, and FFMPEG [10] which conceals whole frame losses using temporal frame interpolation. For the JM decoder, the lost frame is concealed by copying the pixels from the previous frame. For the FFMPEG decoder, a lost P frame is concealed by copying the pixels from the previous reference frame, and a lost B frame is concealed by temporal interpolation between the frame pixels of the previous and the future frames. These two decoders are widely used in academia and industry.

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

We create six different realizations of whole frame loss events of the 20-minute video, producing 900 distinct isolated whole frame losses. All the six lossy videos are decoded by FFMPEG and JM decoders. A subject watches two different loss realizations of whole frame loss events from the same decoder, so a session involves 40 minutes of actual watching time per subject. The experiment takes one hour, including an introductory session and a break. When viewers see a glitch, they respond to that glitch by pressing the space bar. If the response time is within 2 seconds of the loss, the loss is regarded as visible. Each of the 40-minute lossy videos is watched by 10 people. The ground truth loss visibility score for a specific frame loss is calculated as the number of people who see the loss artifact divided by 10. We have a total of 60 people participating in the experiments, where 30 people watch JM-decoded videos and 30 people watch FFMPEG-decoded videos. 1800 ground truth visibility scores are obtained (900 for JM decoder and 900 for FFMPEG decoder).

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

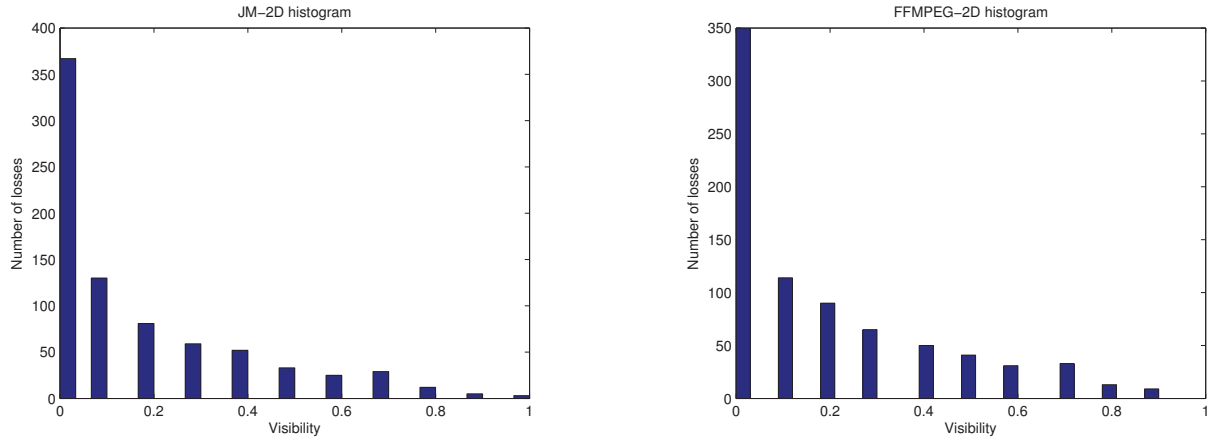
Table 1. Summary of the subjective experiment setup. H is the height of the video.

3. DATA ANALYSIS

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

Figures 1(a) and 1(b) show the histograms of the visibility of the JM decoder and the FFMPEG decoder respectively. For the JM decoder, 40.78% of the losses are not observed by any subjects (visibility is zero). For the FFMPEG decoder, 38.89% of the losses are not observed by any subjects. In other words, more than $1/3$ of losses are not seen by any user. And for the JM decoder, 62.43% of losses have visibility less than or equal to 0.2, whereas for the FFMPEG decoder, 58.29% of losses have visibility less than or equal to 0.2. For both decoders well over half of isolated whole B frame losses are seen by 2 or fewer out of 10 people. One implication is that if we can identify these frames that are less visible to viewers when lost, in the case of network congestion, we can choose to drop unimportant frames to relieve network congestion, and not many end users will observe the losses.

In the design of our experiment, because there is a loss event in every 4 second interval, it could be a concern that viewers would begin to anticipate the next loss event. How-



(a) (b)

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

ever, we do not believe that viewers noticed the loss pattern because there was such a high percentage of loss events which were invisible, so viewers were not perceiving losses in each time slot.

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

Also, we want to know whether one decoder is better than the other in terms of whole frame error concealment visually. We start with a simple paired comparison of the ground truth loss visibility scores between JM and FFMPEG. We say a decoder wins if the ground truth of one decoder is lower (visually better) than the other, and loses if is higher. The result shows that the fractions of JM wins, FFMPEG wins and ties are 33.16%, 29.64% and 37.18%. This means more than $1/3$ of the whole frame losses are observed by exactly the same number of observers for both error concealment methods used. Among the tie cases, 79.05% represent losses with zero visibility for both JM and FFMPEG. Also JM wins more times against FFMPEG. When JM conceals the whole frame loss by frame copy, there are no spatial concealment artifacts; it is just a copy of the previous intact frame. However, for FFMPEG that conceals by temporal interpolation, ghosting artifacts may appear when there is enough motion. A visual example is demonstrated in Figure 3. Frame 35 is lost and concealed by JM with frame copy as in Figure 3(a) and by FFMPEG with temporal frame interpolation as in Figure 3(b). The average whole frame loss visibility over all the data is 0.1716 for JM and 0.1879 for FFMPEG, indicating that on average, the whole frame losses concealed by JM are less visible than by FFMPEG.

For a significance test between the visibility scores of FFMPEG and JM, we can not perform a hypothesis test that assumes the data to be normal (e.g., t test) since from Figures 1(a) and 1(b), their distribution is far from normal. Therefore we resort to nonparametric hypothesis testing. The Wilcoxon Signed Rank Test (paired comparison) [15] compares paired data x and y in a two-sided test where the null hypothesis H_0 is that the median of $x - y$ comes from a continuous, symmetric distribution with zero median, against the alternative that the distribution does not have zero median. Let x_i and y_i be the visibility for FFMPEG and JM in the ith comparison set. Define $w = \sum_{i=1}^{n} r_i z_i$ where r_i is the rank of $|x_i - y_i|$ among all $|x_j - y_j|$, and $z_i = 1$ if $x_i - y_i > 0$ and $z_i = 0$ otherwise. Here $n = 900$, the number of losses. The statistic for the test,

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

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

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

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

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

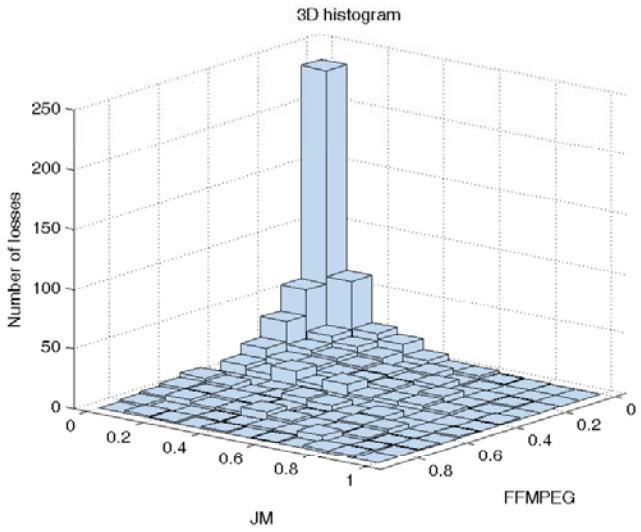


Fig. 2. 3-D Histogram of whole frame loss visibility by JM decoder and FFMPEG decoder

4. WHOLE FRAME PACKET LOSS VISIBILITY MODEL

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

4.1. Factors extractable from bitstream for predicting frame loss visibility

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

Several factors are shown to be important to the prediction of packet loss visibility in our prior study [1, 2]. We consider the residual energy distribution of the MBs in a frame, denoted by **RSENGY**. We take the average of the residual energy of all the MBs in a frame. We denote this quantity as **Mean**RSENGY. **Max**RSENGY denotes the maximal residual energy after motion compensation among all MBs in a frame. **Var**RSENGY denotes the variance of the residual energy of MBs in a frame. Aside from these which were used in [1, 2] , here we include two more descriptions of the distribution. The skewness [15] of RSENGY describes the amount of asymmetry of the RSENGY distribution, denoted as **Skew**RSENGY, and the entropy [16] of RSENGY captures the randomness of the RSENGY distribution, denoted as **Ent**RSENGY.

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

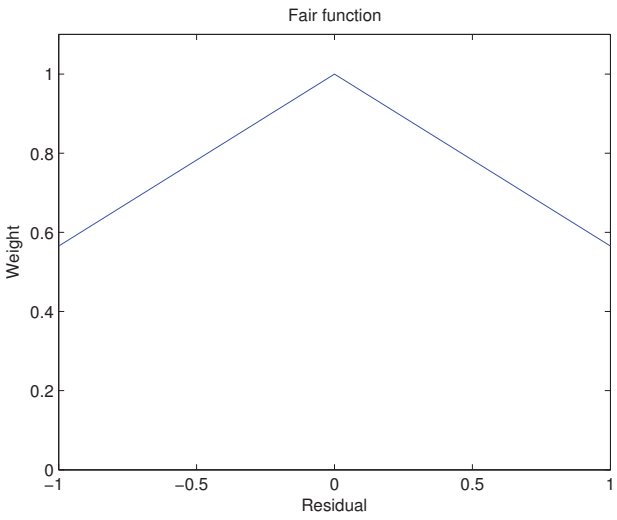


Fig. 4. The Fair function versus the residual.

the phase is well defined. We denote the phase information distribution of the motion vectors as **MotA**. The packet size distribution in bits in a frame, denoted as **SliceSize**, is also included for prediction.

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

For residual energy, as in [1], 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.

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

4.2. Modeling Process

In the experiment and data analysis, we assume each viewer's response is an independent observation of the average viewer (for whom we are developing the model). Therefore, each viewer response can be considered iid with probability p for seeing a particular packet loss. Hence, we choose a generalized linear model (GLM) with the logit function as link function, since it can predict a probability parameter in a binomial distribution.

GLMs are an extension of classical linear models [17, 18]. The probability of visibility is modeled using logistic regression, a type of GLM which is a natural model to predict the parameter p of a binomial distribution [17]. Let $y_1, y_2, ..., y_N$ be a realization of independent random variables $Y_1, Y_2, ..., Y_N$ where Y_i has binomial distribution with parameter p_i. Let $\mathbf{y}$, $\mathbf{Y}$ and $\mathbf{p}$ denote the N-dimensional vectors represented by y_i, Y_i and p_i respectively. The parameter p_i is modeled as a function of P factors. Let $\mathbf{X}$ represent a $N \times P$ matrix, where each row i contains the P factors influencing the corresponding parameter p_i. Let x_{ij} be the elements in $\mathbf{X}$. A generalized linear model can be represented as

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

where $g(.)$ is called the link function, which is typically nonlinear, and $\beta_1, \beta_2, ..., \beta_P$ are the coefficients of the factors. Coefficients β_j and the constant term γ are usually unknown and need to be estimated from the data. For logistic regression, the link function is the logit function, which is the canonical link function for the binomial distribution. The logit function is defined as

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

Often the parameters of the GLM are estimated such that the resulting model has the least deviance (the deviance is a generalization of the residual sum of squares). This treats data points equally, no matter how far they are from the regression line. However, outliers may distort the results. To give unequal treatment to data points to suppress outliers, we minimize the M-estimator [19]; data points farther from the regression line have smaller weights, and contribute less to the final modeling result. We chose the "Fair" function as the M-estimator function, shown in Figure 4. The M-estimator is computed as the sum of the weighted residual squares, where the weight of each data point is computed by the residuals in the previous iteration. The M-estimator function in Figure 4 is chosen to avoid the weights of the curve going close to zero at the two ends, because we do not want to have a final model that has least M-estimator just because most of the data points are at the two ends. The model developing procedure uses 4-fold cross validation to prevent the model overfitting the data, so an average M-estimator is produced for a set of factors.

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

Table 2. Table of factors in the order of importance for Avg_JM_FFMPEG model.

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

Table 3. Table of factors in the order of importance for Max_JM_FFMPEG model.

The factor which most reduces the average M-estimator goes next into the model. This procedure repeats until there is no improvement in the average M-estimator by including an additional factor.

4.3. Discussion

We use the factor set described in Section 4.1, plus interaction terms between any two factors in the set by multiplication between two factors. We then perform the feature selection process described in Section 4.2.

From Section 3 we know that the concealed result for JM is not significantly better than for FFMPEG, and that a whole frame loss with high visibility for one decoder is very likely to be highly visible for the other decoder, therefore it is reasonable to make one generalized model for both decoders. One can make such a model by taking the average of the two visibility scores associated with the same whole frame loss. We denote the result **Avg_JM_FFMPEG**. Another way is taking the maximum of the two visibility scores of the JM and FFMPEG; this aims to predict the visibility for the worst decoder for a loss, and we denote the result **Max_JM_FFMPEG**.

Figures 5(a) and (b) show decreasing M-estimator as we add factors in the order of importance into the models that predict Avg_JM_FFMPEG and Max_JM_FFMPEG respectively. The circle markers in the plots consider all the factors discussed in Section 4.1. We observe that adding more factors in the model produces diminishing returns. In fact, most of those factors involve the computation of skewness and entropy, which are very complicated. Therefore, we remove the factors involving skewness and entropy from consideration. The factors in Figure 5(a) and (b) that are marked by diamonds do not include skewness and entropy. We can see that by saving the computation and reducing the number of factors in the model, we only lose 12.4% for (a) and 6.45% for (b) of the full performance achieved by all the circled factors. The factors in the order of importance and the corresponding coefficients of the final models of Avg_JM_FFMPEG and Max_JM_FFMPEG are listed in Table 2 and Table 3 respectively. One interesting observation is that the first four important factors are the same for both models. Also, for all the factors, the information relating to motion vectors is very important; more than 70% of factors in the model involve motion vector computations. This indicates the amount of motion in the lost frame dominates the visual performance of concealment by both the JM and FFMPEG decoder.

5. CONCLUSIONS

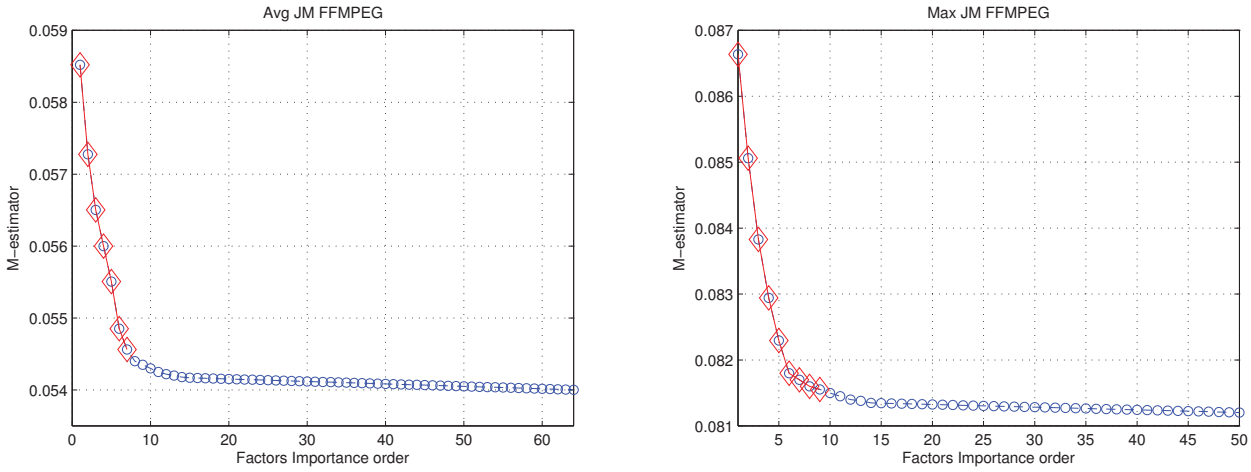
We present a subjective test and its results on whole B frame loss visibility of the H.264 encoded bitstream. We compare the visual result of the concealment by the JM standard and FFMPEG decoders. For whole frame loss, JM produces frame copy artifact, while FFMPEG produces temporal frame interpolation artifact. We found that there is no statistically significant difference in the visibility of these losses between the two different decoders. Experimental results showed that approximately 40% of all isolated losses were not observed by any viewers, and about an additional 20% of the loss events were only observed by 1 or 2 out of 10 observers. We then developed two whole frame loss visibility models; one predicts the average visibility by the decoders, the other is for the worst case visibility .

6. REFERENCES

[1] T.-L. Lin, S. Kanumuri, Y. Zhi, D. Poole, P. Cosman, and A. Reibman. A Versatile Model for Packet Loss Visibility and its Application to Packet Prioritization. *IEEE Transactions on Image Processing*, 19(3):722–735, March 2010.

[2] T.-L. Lin and P. Cosman. Network-based Packet Loss Visibility Model for SDTV and HDTV for H.264 videos. *ICASSP*, 2010.

[3] T.-L. Lin and P. Cosman. Packet dropping for widely varying bit reduction rates using a network-based packet



(a) (b)

Fig. 5. The M-estimator plot versus the number of included factors predicting (a) Avg_JM_FFMPEG (b) Max_JM_FFMPEG.

loss visibility model. *DCC (Data Compression Conference)*, 2010.

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

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

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

[7] VQM. http://www.its.bldrdoc.gov/n3/video/.

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

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

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

[11] R. Pastrana-Vidal, J. Gicquel, C. Colomes, and H. Cherifi. Sporadic Frame Dropping Impact on Quality Perception. *Proceedings of the SPIE Human Vision and Electronic Imaging*, 5292:182–193, 2004.

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

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

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

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

[16] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 2nd edition.

[17] P. McCullagh and J. A. Nelder. Generalized Linear Models 2^{nd} Edition. *Chapman & Hall*, 1989.

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

[19] W. Rey. *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer, 1983.