An Adaptive Error Resilient Video Encoder

Liang Cheng* and Magda El Zarki#
School of ICS and CAL-(IT) 2
UC, Irvine
Irvine, CA 92697
Tel: (949) 824-8584, Fax: (949) 824-4056
*lcheng61@ics.uci.edu, #elzarki@uci.edu

ABSTRACT

When designing an encoder for a real-time video application over a wireless channel, we must take into consideration the unpredictable fluctuation of the quality of the channel and its impact on the transmitted video data. This uncertainty motivates the development of an adaptive video encoding mechanism that can compensate for the infidelity caused either by data loss and/or by the post-processing (error concealment) at the decoder. In this paper, we first explore the major factors that cause quality degradation. We then propose an adaptive progressive replenishment algorithm for a packet loss rate (PLR) feedback enabled system. Assuming the availability of a feedback channel, we discuss a video quality assessment method, which allows the encoder to be aware of the decoder-side perceptual quality. Finally, we present a novel dual-feedback mechanism that guarantees an acceptable level of quality at the receiver side with modest increase in the complexity of the encoder.

Keywords: error propagation, error resilience, motion intensity, feedback, quality measurements

1. INTRODUCTION

The Internet in general, and wireless channels in particular, are best-effort, error prone and lossy environments. Most current video compression standards increase the compression rate by eliminating the spatial and the temporal redundancies. However, the tradeoff of the higher compression rate is its higher susceptibility to errors and the corresponding spatial and temporal propagation of these errors. There are two popular approaches for reconstructing the video at the decoder with improved perceptual quality. One is to conceal the errors at the decoder side by predicting the lost data [6]; the other is to enhance the error resilience of the video at the encoder side [1]-[5]. Both these mechanisms mitigate the loss and improve the perceived quality at the decoder. They are also not mutually exclusive. In this paper, we focus on the design of an error resilient encoder, which is *adaptive* to the losses caused by packet dropping and error propagation).

A variety of techniques have been proposed to enhance the resilience of the video system to the errors. It is widely recognized that intra-coding is an important method to mitigate the effect of an error in a predictive video compression system. A large body of literature proposes different heuristic refreshing strategies, including the periodical refreshing of the whole frame, progressive refreshing, and conditional replenishment. Frame level refreshing is the most reliable mechanism to refresh the data and prevent the past errors from propagating to successive frames. However, for low bit-rate channels such as most wireless channels, transmission of an intra-frame may cause delay and/or lower the quality of the subsequent frames when using rate control strategies [10]. Traditional progressive intra-coding can avoid the sudden spike of the number of output bits. However, the frequency of the refreshing is still uncertain, even though some studies have empirically related the refreshing rate to the packet loss rate [17]. The conditional replenishment methods [4][5][12] selectively update the macroblocks based on the notion that the important areas must be updated more frequently, e.g., the adaptive intra refreshing (AIR)[12], applies the frequent intra-updates to the regions undergoing significant changes. However, for critical applications, such as remote medical diagnosis and remote

Visual Communications and Image Processing 2003, Touradj Ebrahimi, Thomas Sikora, Editors, Proceedings of SPIE Vol. 5150 (2003) © 2003 SPIE · 0277-786X/03/$15.00

surveillance, etc., the long-term inaccuracy of some regions, even if they do not experience much change, may decrease the fidelity of the whole monitoring system.

To achieve the optimal/sub-optimal inter/intra macroblock mode selection, many recent papers consider incorporating the error concealment method in the computation of the decoder distortion at the encoder. They outline mode selection based on an end-to-end rate-distortion framework [1]-[3]. In these papers, minimizing the distortion between the original video and the decoder-side reconstructed video is the main goal. The encoder is required to repeat the decoding process and thus able to know in advance the macroblock affected by the error propagation given the feedback of packet loss rate or the specific lost macroblock location.

However, this coupling of the encoding process to the error concealment at the decoder violates the independence between the encoder and the decoder. It converts the mapping relationship between the encoder and the decoder from a "one-to-many" to a "one-to-one", which limits the flexibility of the encoder and the versatility of the decoder. Even with the assumption that the encoder can handle the complexity added by incorporating every possible error-concealment algorithm, the uneven delay of the loss information delivered to the encoder may render the error prediction inaccurate. In addition, these approaches are difficult to be applied in the multicast scenario, because their mode switching solution is unique, i.e., the claimed optimal/sub-optimal result.

After studying the merits and demerits of the above error resilience techniques, we propose a quality guaranteed progressive replenishment method, which jointly considers the nature of temporal error propagation, the packet loss rate and the possible variation of error concealment algorithms in different decoders. Our contribution is composed of two parts - *continuous progressive replenishment* and *quality-conditional progressive replenishment.* The former method is built on the assumption that no (or very poor) error concealment exists at the decoder, and the refreshing rate of the encoder is adapted to the video motion intensity and the packet loss rate only. We demonstrate our result by comparing it to the well-known *AIR* approach, which has been adopted by the MPEG-4 standard [19]. The quality-conditional progressive replenishment method incorporates a perceptual quality feedback channel that enables the encoder to assess the perceived quality at the decoder. Instead of continuously using adaptive progressive replenishment, we enable progressive refreshment for a short duration only when the perceived quality drops below a specified threshold at the receiver. The quality awareness is achieved by comparing the extracted quality features of the decoded video at the receiver to that of the original video at the encoder. The features are transmitted to the encoder via a low bandwidth feedback channel [13]-[15]. Hence, we are able to guarantee the predetermined minimum video perceptual quality in our system.

The rest of this paper is organized as follows. In the next section, we talk about the factors that may cause or aggravate the errors. In section 3, we describe our proposed adaptive *continuous progressive replenishment* scheme, which we compare to the performance of AIR. In section 4, we discuss an in-service video quality assessment system [13]-[15] and extend the *continuous progressive replenishment* to a quality-enabled system by incorporating the perceptual quality feedback information in the replenishment scheme.

2. ERROR FACTORS

2.1 Packet loss

The best-effort nature of the Internet, can lead to high packet loss rates. The intermediate nodes of the network usually discard packets due to buffer overflow, long queuing delays, etc. With the current infrastructure, it is difficult to prevent packet loss from occurring and to predict the location of the next lost packet. However, we can use packet loss rate (PLR) information to estimate the degree of network congestion. For wireless systems, the loss rate does not necessarily indicate congestion, as the channel can experience severe fluctuations in quality that can lead to bit errors resulting in packet losses. The PLR in that case can indicate channel status (long fade situations).

2.2 Error propagation

An error caused by a packet loss is generally exacerbated for predictive video coding systems, because of the prediction loop. Error propagation is usually composed of both spatial propagation and temporal propagation. The spatial propagation caused by spatial prediction (AC/DC prediction, motion vector spatial prediction), can be effectively confined within the packet boundary. For those compression standards that use inter-frame prediction with motion estimation/compensation, the temporal error propagation is inevitable and difficult to be suppressed. At the encoder side, the inter-mode macroblock is first predicted from the previous decoded frame via motion estimation. Then the most matching region is found, as seen on the left part of Figure 1. The prediction residue is transform-coded, quantized, run-length coded, converted to variable-length-code (VLC), and then packetized for transmission. At the decoder side, if all or part of the matching region is lost, the macroblock that utilizes it as a reference will be affected during the motion compensation phase, as seen on the right part of Figure 1.

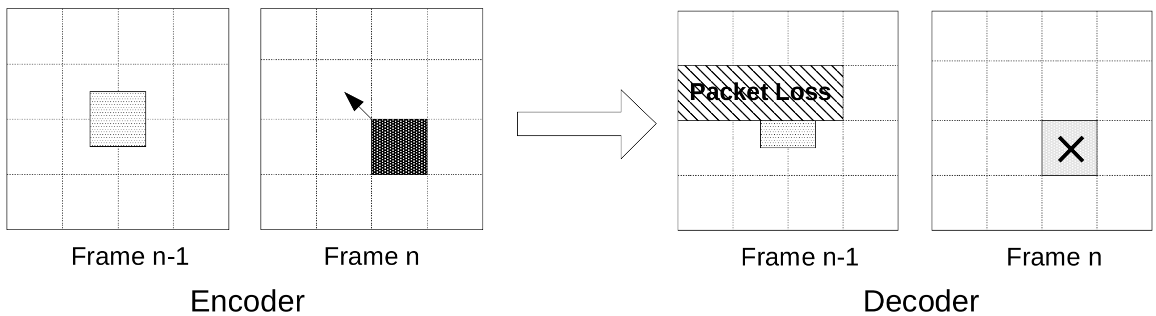


Figure 1. Temporal Error propagation caused by motion estimation/compensation

2.3 Concealment variation at the decoder

Error concealment implemented at the decoder, can improve the quality of the received video. It uses the redundancy of the compressed data to predict the lost data, such as motion vector, DCT coefficients, etc. The basic role of an error concealment scheme is packet loss detection and re-synchronization. Another important task, namely, loss prediction, is highly dependent on the video content. For example, for a continuous video scene with slow motion, the quality of error concealment can be high. The more sophisticated the concealment algorithm the better quality of the displayed video at the receiver. Of course, this utilizes the cost of more CPU cycles and memory, which may lower the decoding speed.

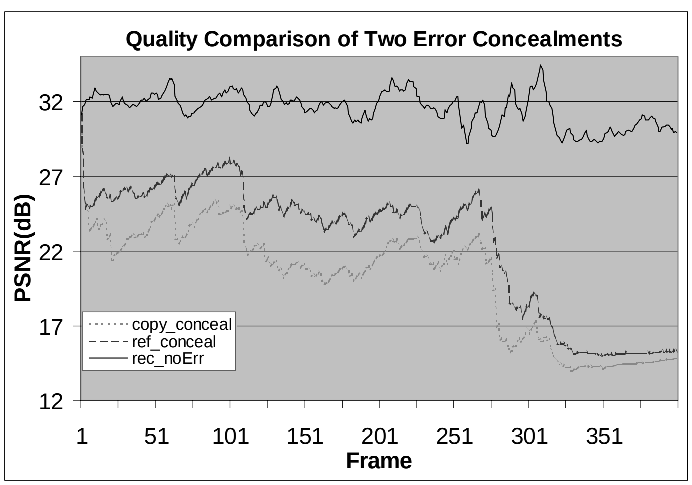


Figure 2. Different error concealments give various recovered quality (*Foreman;* 64 kbps)

Figure 2 shows the recovered quality of different decoders (MPEG-4 decoder with different error concealment algorithms.) We first show the simplest concealment method, i.e., simple macroblock replacement from previous frame with no motion compensation (motion vector is zero.) We then implement a better solution that uses the motion vector of the reference frame for motion compensation. Obviously, none of these two error concealments can completely recover the lost data. However, since the second solution utilizes the redundancy of the motion vector, it can recover data better than the first one. Therefore, the recovery capability of the decoder, i.e., error concealment algorithm implemented, should have an impact on the error resilience capability of the encoder.

3. ADAPTIVE PROGRESSIVE REPLENISHMENT

In this section, we present an adaptive progressive replenishment scheme that overcomes network losses and the corresponding temporal error propagation. This scheme assumes that the encoder has received no decoder information. We hold the notion that the refreshing rate should catch up with the packet loss rate and the corresponding error propagation speed.

3.1 Motion intensity measurement

From the above discussion, we conclude that the motion compensation is the reason of the temporal error propagation. The overall motion of the scene directly determines the overall error spreading speed, as seen in Figure 3. We thus conceptually link our progressive refreshing scheme to the motion intensity.

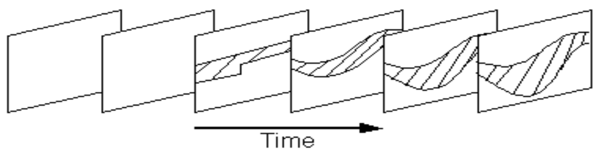
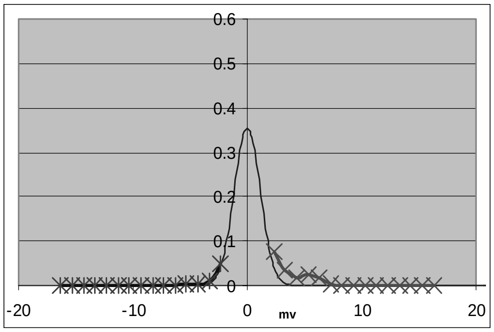


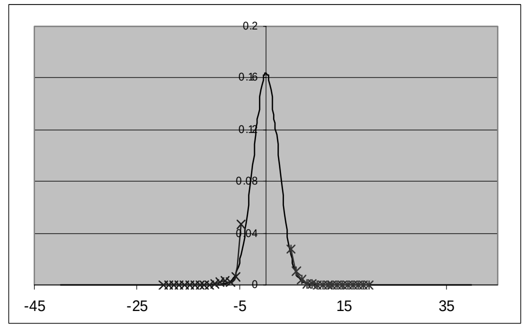
Figure 3. Error spreads based on the scene motion

In the study below, our analysis and figures will be focusing on the horizontal component of the motion vector (henceforth referred to MVX) as we progressively refresh the frame, column by column. The reason of choosing a column-by-column refreshing pattern is that most of the macroblock level rate control mechanisms don't consider the macroblock type [11]. The rate control mechanism and the packetization scheme are generally left to right and row-by-row. The buffer status used by the rate control scheme determines the quantization parameter being applied to the next macroblock [7]-[9] without checking its mode. Thus, if many intra-macroblocks are packed in one packet, the rate control mechanism most likely will assign a lower quantization parameter to the macroblocks before that packet and a higher one for the macroblocks after it, which will result in an inconsistent quality for the frame. One solution to overcome this quality inequity is to refresh the frame on a row-by-row basis and add different weights to the quantization parameter of the macroblocks before and after the "intra" packet. The other solution, which is used in this paper, is on a column-by-column basis, which maximally distributes the intra-macroblocks to different packets. Hence, there is no need to change the existing rate control algorithm to avoid the quality inequity.

From empirical tests, we found that the percentage of zero valued motion vectors is high regardless of how motion intensive the scene may be. If we remove the zero valued motion vectors and eliminate their influence, we find that the distribution of the non-zero valued motion vectors follows the tails of a Gaussian distribution. The standard deviation of the non-zero valued motion vectors will be used to define the shape of the bell. The area of the Gaussian "bell" matches the percentage of the zero valued motion vectors. In Figure 4, we compare the distribution of the non-zero valued motion vectors to the standard Gaussian distribution with the same standard deviation after shifting the positive non-motion vector and negative non-motion vector forward and backward, respectively.



(a) *Foreman*, Z = 74.76%, σ = 1.131



(b) *Suzie*, Z = 89.04 % σ = 2.438

Figure 4. Distribution of non-zero valued motion vectors matches the tails of the Gaussian Distribution (abscissa denotes shifted motion vector value; ordinate denotes percentile)

Based on the above analysis, we derive equation 1 below to represent the motion intensity m of every frame. To describe our scheme, we need to define a set of variables.

Z : Cumulative probability of zero value motion vector, e.g. Z = 89.04% for the video sequence *Suzie* and 74.76% for the video sequence *Foreman*.

σ : Standard deviation, obtained from the non-zero value motion vectors.

E : Mean of the non-zero value motion vectors.

p : Percentage of motion vectors values that are less than a threshold.

I : Interval between two frames that incur MB column refreshment.

$$m = \text{Max}\ \{((\Phi(\frac{1+p}{2}) - \Phi(\frac{1+Z}{2})) \times \sigma + 1 + E) / 16, 0\}, \qquad (1)$$

where Φ (.) is the inverse Gaussian distribution.

Our refreshing direction is column-by-column, from left to right, so we only consider the positive values of m . This means that the error propagation in the leftward direction is not considered. Our tests show that the dual-directional refreshing approach (i.e., both leftward and rightward) lowers the coding efficiency with only slight error resilience gain compared to the unidirectional method that we adopted for the remainder of this paper.

In equation 1, p should be chosen at least larger than the percentage of zero value motion vectors. However, p cannot be so large that the largest motion vector is always picked. The optimal value of p changes according to the content of the video. We found that p equal to 99% gives consistently good results for all video sequences that we tested.

3.2 Motion adaptive force-update macroblock (MAFUMB)

Based on the motion intensity calculated by equation 1, we developed the following algorithm to determine the macroblock-column refreshing rate.

Step 1:

After motion searching, process the obtained motion vectors; obtain their statistics – average and standard deviation and the percentage of zero value motion vectors.

Step 2:

After P-frame coding, calculate m by using equation 1. Since the motion vector value is normally clipped [-16, 16]. m is less or equal to 1, $\left\lfloor \frac{1}{m} \right\rfloor$ is used to represent the frame intervals to be skipped before the next column refreshing.

In case of unrestricted motion search, m may be greater than 1. Then the number of macroblock columns to be refreshed is set to $\lfloor m+1 \rfloor$. (Note: $\left\lfloor \frac{1}{m} \right\rfloor$ is clipped to [0, 100]).

The dark-color area represents m of every frame.

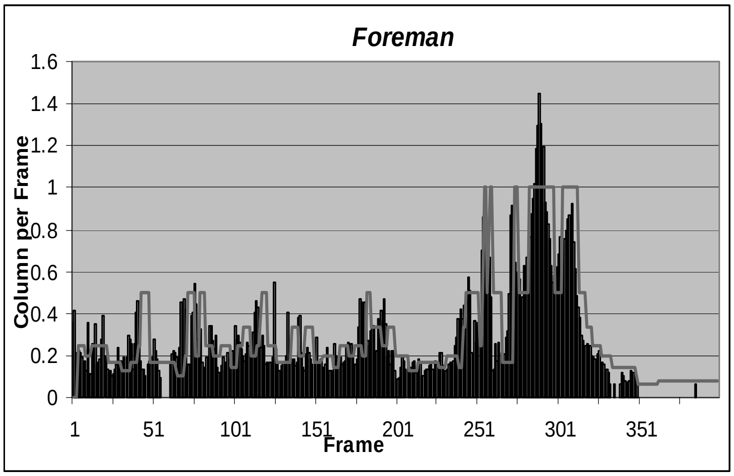


Figure 5. Adaptive refreshing based on motion intensity (the dark-color area is the motion intensity of every frame; the continuous curve represents the

Figure 5 shows the calculated motion intensity and the refreshing rate. The most motion intensive scene occurs in the last part of the clip, when the camera pans. Based on our analysis, the error propagates faster during that period, the refreshing rate is therefore correspondingly more frequent. Figure 6 compares the decoding result using the MAFUMB scheme and the standard MPEG-4 encoder. The packet loss rate is 0.5%, which is based on the Gilbert-Elliott packet erasure model (see Figure 8). The error concealment algorithm used by the decoder consists of simple copying, as explained in section 2.3. The MAFUMB scheme appropriately enhances the error resilience by considering the temporal error propagation, as shown in Figure 6.

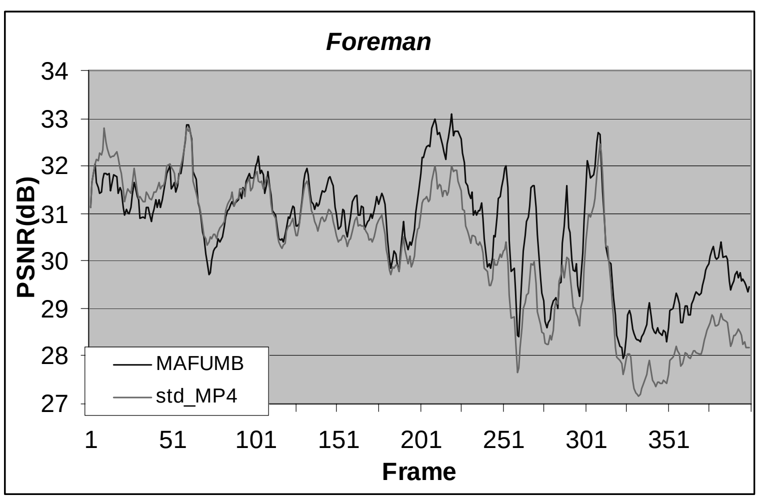


Figure 6. MAFUMB vs. standard MPEG-4 encoder

3.3 MAFUMB with PLR feedback

For our scheme, we assume a uniform packet loss distribution (the worst-case scenario as shown in Table 2) to provide an upper bound on the error resilience ability of our scheme. We assume that the decoder can only provide re-synchronization to continue the decoding without using any error concealment algorithm to predict the lost motion vectors or other lost information.

The concept of our algorithm is that the refreshing rate should be fast enough to refresh the whole frame in error before the next error in a frame occurs, and thus the latency of the error cannot exceed the interval between two packet losses. We therefore augment the refreshing scheme based on equation 1 by including packet loss information in the refreshing update formula, as given by equation 3 below.

To describe our scheme, we need to define a set of variables.

R_{final} : Final adaptive refreshing rate, which is determined on both the motion intensity m and PLR.

R_{PLR} : Adaptive refreshing rate, which only depends on PLR.

H : Number of macroblock columns in one frame.

P_{loss} : The PLR obtained from feedback reports such as those provided by RTCP

B_{rate} : The bit rate of the video stream

L_p : The packet length

R_{frame} : The frame rate

We propose the following heuristic solution:

$$R_{PLR} = H \times \frac{B_{rate}}{R_{frame} \times L_p} \times P_{loss}\,; \qquad (2)$$

$$R_{final} = H \times \frac{B_{rate}}{R_{frame} \times L_p} \times P_{loss} + \frac{1}{I}\,; \qquad (3)$$

Based on equation 2, we calculate the number of columns per frame that need to be refreshed to adapt to the packet loss rate, the results are shown in Table 1.

Packet Loss Rate (%)	R_{PLR} (Column Refresh / Frame interval)
1	0.3
5	1.5
10	3
20	6

Table 1. Progressive refreshing rate based on packet loss rate

In Figure 7, we plot the video quality loss (based on equation 4) for three typical channels: all errors in one burst, uniformly distributed error model, and Gilbert-Elliott packet erasure model with no burst. As seen in Figure 8, the Gilbert-Elliott packet erasure model uses a Markov model with 2 states, denoted as G (Good state) and B (Bad state), to simulate the packet drop due to the network congestion. r represents the probability with which the network transits from uncongested state to congested state. q is the probability of transiting from congested state to uncongested state. Thus, $\frac{1}{q}$ can be derived to be the burst length, namely the length of consecutive lost packets: $r = \frac{PLR \times q}{1 - PLR}$. In our simulation, the result based on the Gilbert-Elliott model is obtained by averaging 30 trials. Table 2 demonstrates that the uniform distribution gives the worst-case results, which satisfies our assumption of the worst-case analysis.

$$\Delta PSNR = PSNR_{dec} - PSNR_{enc} \quad (4)$$

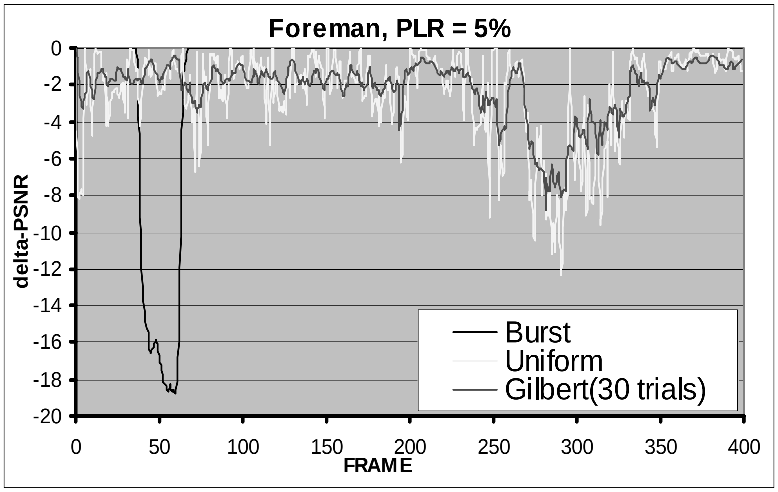


Figure 7. Results based on three distributions, which all correspond to a 5% packet loss rate as reported by RTCP.

Distribution	Burst	Uniform	Gilbert
$\Delta PSNR$ (dB)	-1.045	**-2.463**	-2.170

Table 2. Impact of different error distributions

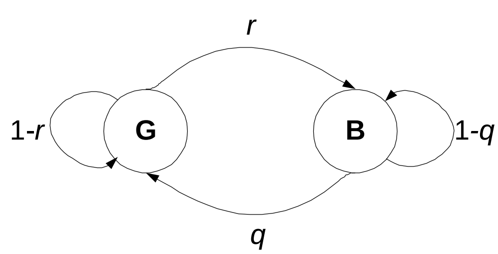


Figure 8. Gilbert Elliott packet erasure channel model

3.4 Packet loss adaptive MAFUMB vs. AIR

In this section, we compare our packet loss adaptive MAFUMB to the AIR scheme. Since AIR does not use feedback information, we encode the *Foreman* sequence using AIR with different intra-macroblock numbers, i.e., 2, 6, 10, 20, 30, 40 and 60. Each of these encoded streams is decoded by using simple copying error concealment and transmission over a Gilbert-Elliott packet erasure channel. In Figure 9, the quality results based on AIR are shown with dots. The results using adaptive MAFUMB are shown with a line. We can conclude from Figure 9 that adaptive MAFUMB is consistently superior to AIR.

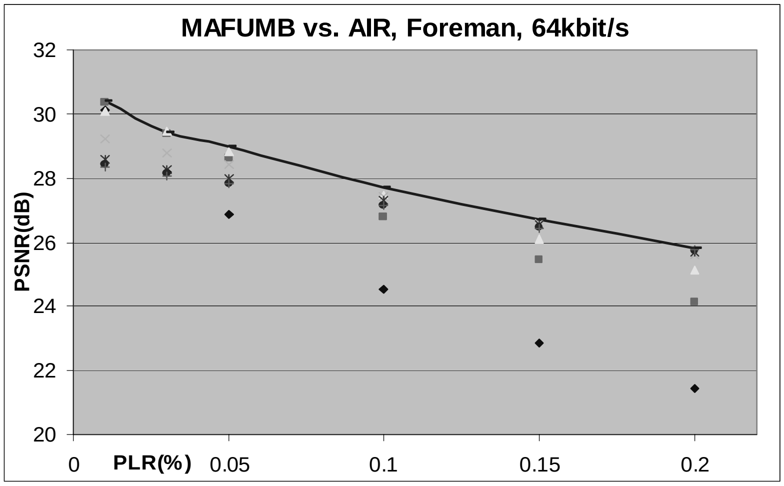


Figure 9. Packet loss feedback MAFUMB vs. AIR

4. QUALITY FEEDBACK ADAPTATION

Video quality assessment is the most reliable and direct way to measure the quality of service at the receiver side, because the ultimate aim of video encoding is to allow the human to receive perceptually acceptable reconstructed video after decoding. Two challenges arise: (1) How to measure the video quality? (2) How to monitor the decoder constructed video quality at the encoder side in a real-time mode?

4.1 Video quality measurement

Objective quality assessment mathematically compares the original video and the reconstructed video on a pixel-by-pixel basis. Peak signal noise ratio (PSNR) is the most commonly used equation to obtain the objective quality. Most objective measurements, such as PSNR, are implemented off line, and thus cannot be used for in-service monitoring.

The authors in [14] propose a perceptual quality scheme for in-service digital video quality monitoring. Instead of using pixel-comparison, the ITS model extracts the features from the processed spatial-temporal (S-T) regions of both the original and reconstructed video streams. The feedback rate for the quality features from the decoder is extraordinarily low, e.g. for quarter-common-interchangeable-format (QCIF) video, the feedback rate is only 15.47 kbit/s. In addition, the quality result of the ITS model highly correlates with subjective measurement results [14].

4.2 Dual-feedback design

In the sections above we described a mechanism by which we can adjust the refresh rate based on motion intensity and the PLR. However, it is not always obvious how packet loss has impacted the received quality, in particular since different error concealment algorithms can give highly varying results. We therefore propose to use the perceptual quality feedback to monitor the quality of the decoded video and use that to determine the refresh rate. Therefore, instead of using equation 3 continuously upon any packet loss report, we only adapt the refreshing rate to the potential error propagation rate (based on motion intensity) when the perceived quality drops below a threshold. When the quality drops below an acceptable threshold, the encoder uses adaptive MAFUMB with the latest packet loss rate to increase the error resilience capability of the successive compressed frame. Figure 10 describes the two feedback mechanisms in our system. The packet loss rate is fed back via an RTCP channel, and the quality feedback uses a low bit rate TCP channel.

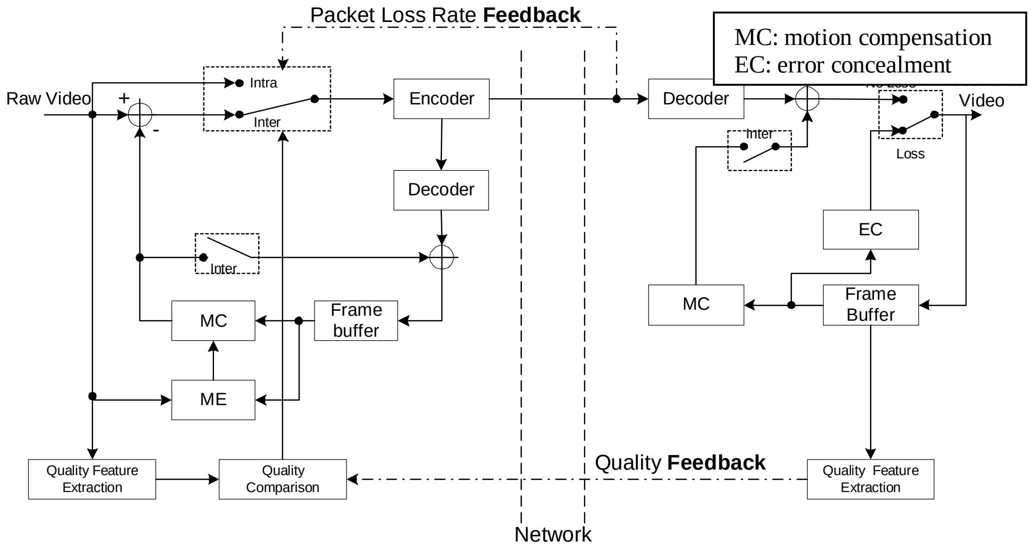
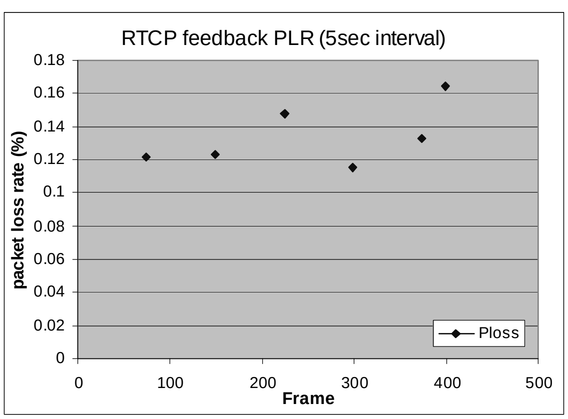


Figure 10. The proposed end-to-end dual-feedback error control system**.**

4.3 Demonstration of the dual feedback encoder

The implementation is carried out by modifying the MPEG-4 natural video encoder/decoder [18]. We compressed 400 frames from the *Foreman* QCIF video sequence. The error environment is simulated by using an error pattern file used in [16] for describing a wireless channel. The transmission protocol stack is designed as RTP/RTCP/UDP/IP. We use the ITS model to evaluate the quality for the simulation involving perceptual quality-feedback.

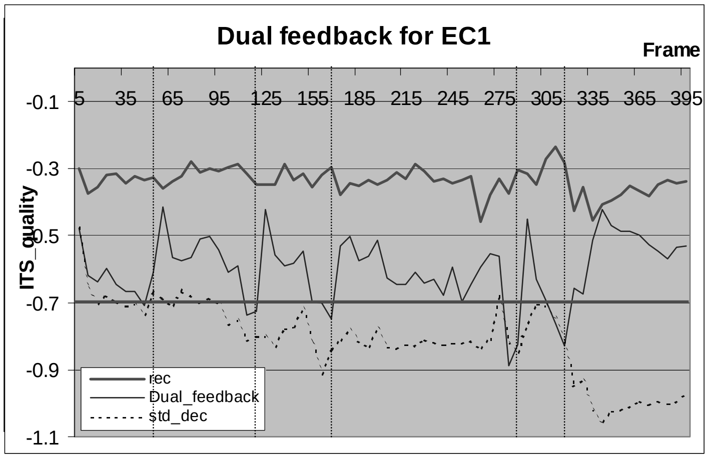
We use the MPEG-4 video packetization scheme, which packs approximately an equal number of bits into every packet on the macroblock boundary. The additional overhead for transmission, i.e., IP overhead, RTP overhead, UDP overhead, comes to 40 bytes. The checksum is used for error discovery. No error correction is implemented, i.e., we drop a packet whenever there are bit errors. Figure 11 shows the RTCP feedback report based on the error pattern file from [16].



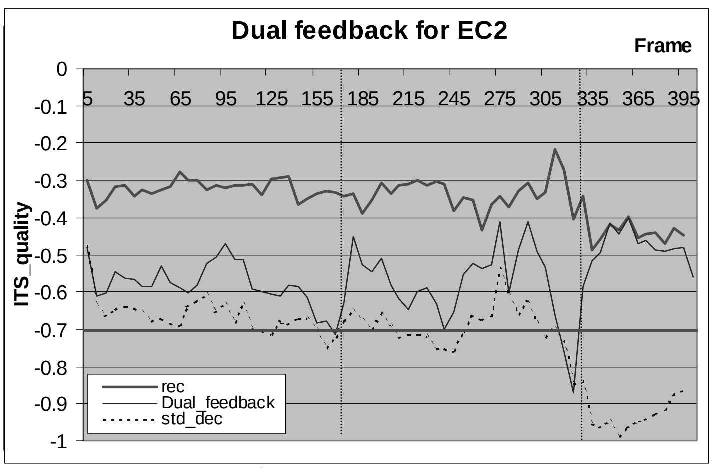
Error File: wcdma-64kb-070hz-3
Document: Q11-F-05
User bit-rate: 64 kb/s
Carrier frequency: 1.9 GHz
Average bit-error rate: $1.26*10^{-3}$
Average burst length(bits): 17
Doppler frequency: 70 Hz (i.e. 40km/h)
Frame rate: 15 fps
RTCP interval: 5 sec

Figure 11. RTCP feedback simulation of the error pattern file.

With the aid of perceptual quality feedback and the packet loss rate feedback, we are able to achieve the quality guarantee. The quality of the recovered video at the decoder basically fluctuates between the reconstructed quality measured at the encoder side and the specified quality threshold, which is set to –0.7. When comparing Figure 12(A) and Figure 12(B), we can also conclude that our approach treats decoders with various error concealment algorithms differently, adapting to their ability to recover from errors. Our approach increases the quality of the recovered video much better than the standard MPEG-4 encoder.



(A) EC1 denotes "direct copy" error concealment. Quality drop responses at frame #53, #119, #167, #287, #317, and #329.



(B) EC2 denotes "using reference motion vectors" error concealment. Quality drop responses at frame #173 and #323.

Figure 12. Quality guarantee demonstration for dual-feedback error control

5. CONCLUSIONS

In this work, we presented a real-time encoder-side error resilience tool to mitigate the error caused by different sources - packet loss, motion intensity and error propagation, and the decoder error concealment variation. Based on feedback of the packet loss rate and the motion intensity analysis, our approach (adaptive MAFUMB) is capable of providing better quality than the AIR scheme. In addition, we propose an additional low-bandwidth channel for the perceptual quality feedback, which can be used to trigger the adaptive MAFUMB. The simulation results demonstrate our dual feedback scheme is capable of *nearly* guarantying the quality of different decoders. The focus of our future work is to (1) investigate the selection of the lowest satisfactory quality bound; (2) optimize the inter/intra switching of encoder in conjunction with quality feedback.

6. ACKNOWLEDGEMENTS

We would like to thank Mr. Xiaoxiang Lu for his collaboration in this paper.

7. REFERENCE

1 R. Zhang, S. L. Regunathan and K. Rose, "Video Coding with Optimal Inter/Intra Mode Switching for Packet Loss Resilience," *IEEE Journal on Selected Areas in Communications, Special Issue on Error-Resilient Image and Video Transmission*. pp. 966-976, vol. 18, no. 6, June 2000.

2 D. Wu, T. Hou, B. Li, W. Zhu, Y.-Q. Zhang, H. J. Chao, "An End-to-End Approach for Optimal Mode Selection in Internet Video Communication: Theory and Application," *IEEE Journal on Selected Areas in Communications (JSAC)*, Special Issue on Error-Resilient Image and Video Transmission, vol. 18, no. 6, pp. 977-995, June 2000.

3 G. Cote, S. Shirani, and F. Kossentini, "Optimal Mode Selection and Synchronization for Robust Video Communications over Error-Prone Networks," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 18, no. 6, pp. 952 – 965, June 2000.

4 P. Haskell and D. Messerschmitt, "Resynchronization of motion compensated video affected by ATM cell loss," *Proc. 1992 Int. Conf. Acoust., Speech, Signal Processing*, San Francisco, CA, Mar. 1992, pp. 545-548.

5 S.T. Worrall, A.H. Sadka, P. Sweeney, A.M. Kondoz, "Motion Adaptive Error Resilient Encoding for MPEG-4," *IEEE ICASSP 2001*, Vol. 3, pp. 1389-1392, May 2001.

6 Y. Wang and Q. F. Zhu, "Error control and concealment for video communication: A review," *Proc. IEEE*, vol. 86, pp. 974 – 997, May 1998.

7 Test Model Editing Committee, "MPEG-2 Video Test Model 5," ISO/IEC JTC1/SC29/WG11 MPEG93/457, April 1993.

8 ITU study group 16, Video Coding Experts Group, "Video Codec Test Model, Near Term, Version 8 (TMN8), Revision 1," Q15-B-46, Sunriver, 8-11 September 1997.

9 ITU study group 16, Video Coding Experts Group, "Video Codec Test Model, Near-Term, Version 10 (TMN10), Draft 1," Q15-D-65d1, Tampere, Finland, 21-24 April, 1998.

10 Supavadee Aramvith, I-Ming Pao, and Ming-Ting Sun, "A Rate-Control Scheme for Video Transport over Wireless Channel," *IEEE Trans. on Circuit and Systems for Video Technology*, vol. 11, no. 5, pp.569 - 580, May 2001.

11 Jeong-Woo Lee, Doe-Hee Kim, and Yo-Sung Ho, "A Rate Control Algorithm using Macroblock Types," *Proceedings of IEEE TEMCPM – Speech and Image Technologies for Computing and Telecommunication*, 1997.

12 Information technology – Coding of audio-visual objects – Part 2: Visual, ISO/IEC 14496-2: 1999/FDAM 1: 1999(E).

13 Xiaoxiang Lu, Ramon Orlando Morando, Magda El Zarki, "Understanding Video Quality and Its Use in Feedback Control," *International Packet Video Workshop*, Pittsburgh, PA, USA 2002.

14 Stephen Wolf, and Margaret H. Pinson, "Spatial-temporal distortion metrics for in-service quality monitoring of any digital video system", *Proc. of SPIE International Symposium on Voice, Video, and Data Communications,* Boston, MA, September 11-22, 1999.

15 X. Lu, S. Tao, M. El Zarki, and R. Guerin, "Quality-based Adaptive Video Over the Internet," *Proceedings of CNDS '2003*, Orlando, FL, January 2003.

16 ITU study group 16, CSN, Multimedia Terminals and System Experts Group, "WCDMA Error Patterns at 64 kb/s," Cannes, June 8-11, 1998.

17 G. Cote, F. Kossentini, "Optimal Intra Coding of Blocks for Robust Video Communication Over the Internet," *Signal Proc.: Image Communication*, vol. 15, pp. 25-34, 1999.

18 Information technology – Coding of audio-visual objects – Part 5, Video Reference Software, ISO/IEC 14496-2, Version: *Microsoft-FPDAM1-1.0-000403*.

19 Information technology – Coding of audio-visual objects – Part 2: Visual, ISO/IEC 14496-2: 1999/FDAM 1: 1999(E).