Content-Aware Error Resilient Transcoding Using Prioritized Intra-Refresh for Video Streaming

Hung-Jyh Chiou, Yuh-Ruey Lee, and Chia-Wen Lin

Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan

Submitted to the Special Issue on **Visual Communication in the Ubiquitous Era**
Journal of Visual Communication and Image Representation

Submitted, March 15, 2004
Revised, July 18, 2004
Accepted, November 20, 2004

Corresponding Author:
Prof. Chia-Wen Lin
Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan
Phone: 886-5-272-0411 ext. 33120
Fax: 886-5-272-0859
Email: cwlin@cs.ccu.edu.tw

Biographies for Authors

Hung-Jyh Chiou received the B.S. degree from Feng-Chia University, Taichung, Taiwan, in 2001, the M.S. degree from National Chung Cheng University, Chiayi, Taiwan, in 2003. Since September 2003, he has been with Institute for Information Industry, Taipei, Taiwan, as a software engineer. His research interests include digital video coding and multimedia communications.

Yuh-Ruey Lee received the B.S. degree in fiber science and engineering from National Taiwan University of Science and Technology University, Taipei, Taiwan, in 1994, the M.S. degree in computer science from National Chung Cheng University, Chiayi, Taiwan, in 2003. Since 2002, he has been working toward the Ph.D. degree in Department of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. His current interests include video coding and video content adaptation.

Chia-Wen Lin received the M.S. and Ph.D. degrees in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, R.O.C., in 1992 and 2000, respectively. In August 2000, he joined the Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan, R.O.C., where he is currently an Assistant Professor. Prior to joining National Chung Cheng University, he was a Section Manager with Computer and Communications Research Laboratories (CCL), Industrial Technology Research Institute (ITRI), Taiwan, R.O.C. His research interests include video coding and networked multimedia technologies.

Dr. Lin was the recipient of the 2000 Research Achievement Award presented by ITRI. He was also the recipient of the 2000 and 2001 Best Ph.D. Thesis Awards presented by the Acer Foundation and the Ministry of Education, R.O.C., respectively.

Content-Aware Error Resilient Transcoding Using Prioritized Intra-Refresh for Video Streaming

Hung-Jyh Chiou, Yuh-Ruey Lee, and Chia-Wen Lin[1]

Department of Computer Science and Information Engineering
National Chung Cheng University
Chiayi 621, Taiwan

Abstract -- Transmitting video data over wireless networks can be very unreliable due to packet-loss, leading to serious video quality degradation which is annoying to human perception. The lost packets not only affect the quality of current frame, but also lead to error propagation to subsequent frames due to motion-compensated prediction techniques used in standard video codecs. Adding error-resilience to video bitstream for robust video delivery to users thus becomes a very important issue. In this work, we propose a two-pass intra-refresh transcoding scheme for inserting error-resilience features to a compressed video at the media gateway of a three-tier streaming system. The proposed transcoder can adaptively vary the intra-refresh rate according to the video content and the channel's packet-loss rate to protect the most important macroblocks (MBs) against packet loss. In the first-pass encoding, the encoder estimates the amount of error propagation at MB level, and then generates side information as transcoding hints for use at the transcoder. In the second-pass transcoding, the error-resilient transcoder adaptively determines the intra-refresh rate and the locations of MBs to perform intra-refresh according to the side information. Experimental results show that the proposed method can effectively mitigate the error propagation due to packet loss so as to improve the visual quality significantly. The proposed transcoder can also meet the realtime requirement.

Keywords-- video coding, video transcoding, error resilient coding, intra refresh, video streaming.

[1] Corresponding Author

1. Introduction

With the rapidly growing demand and widely deployed infrastructure of wireless networks (e.g., GPRS, 3G, and wireless LANs), applications of streaming video over wireless links have attracted much attention in recent years. However, the packet erasure and bandwidth variation characteristics of wireless links still present a number of challenges to streaming video applications [1]-[3]. In a video streaming system, a server pre-stores encoded video streams and transmits them to client terminals for decoding and playback. There are several existing video coding techniques, for example, H.26x, and MPEG-x, developed to compress video sequences into bitstreams to reduce the data sizes. These video encoding techniques exploit spatial and temporal redundancy to achieve a high compression ratio, while making the compressed data very sensitive to transmission error [3]. Video transport over wireless networks may suffer from signal fading and network congestion which will cause packet loss/erasure. This packet-loss problem may lead to serious video quality degradation, which not only affects the quality of current frame, but also leads to error propagation to subsequent frames due to the motion-compensated prediction technique used in standard video codecs [3]. Furthermore, in practical applications that video contents are compressed and stored for future delivery, the encoding process is typically performed without enough prior knowledge about the channel characteristics of network hops between the encoder and the decoder. In addition, the heterogeneity of client networks also makes the encoder very difficult to adapt the video contents to a wide degree of different client channel conditions, especially for wireless client terminals. In order to achieve error robustness for transmitting video over wireless networks, the server must be able to adapt or transcode the non-error-resilient compressed video streams into error-resilience-capable streams at the intermediate network node. To serve this purpose, a video transcoder [4-9] can be placed in a network node (e.g., mobile switch/base-station, proxy server, and video gateway) connected to a high-loss network (e.g., wireless network or highly congested network) to insert error-resilience features into the video bitstream to achieve robust video transmission over wireless channels [10-15].

Fig. 1 illustrates a three-tier video streaming system for home networking. This application scenario involves a streaming server, a media gateway (e.g., home server), and a number of client terminals (e.g., information appliances). In a home network, the communication links to client terminals may have different packet-loss rates and channel bandwidths for different clients. The home server has to deploy different error-resilience features and regulate the bitrate in order to match different channel characteristics. A transcoder is usually located at the home server for adapting the incoming video bitstream to the varying channel conditions. Using a transcoder to handle the different demands (e.g., bandwidth, resolution, frame-rate, and channel condition) from different client devices can reduce the complexity and transmission cost from the streaming server to the home receivers. A typical example of error-resilient transcoder with feedback is shown in Fig. 2 [1,11,12]. The transcoder first extracts the video features (e.g., locations of video data which are likely to result in more serious error propagation if lost) from the incoming bitstream as well as estimates the client channel conditions according to the feedback channel statistics. The extracted features and the estimated channel condition are then used to determine the error-resilient coding policy for guiding the joint allocation of source/channel coding resources. The features of video contents can also be pre-computed in the front-end encoding process and sent to the transcoder as auxiliary data (metadata) to assist the transcoding. In this work, we investigate efficient error-resilient transcoding methods for such the three-tier architecture with the transcoder located at the home server for enhancing error robustness to video streams prior to delivering video data to the mobile users.

Commonly used error resilient source coding tools include data partitioning, synchronization marker, RVLC (Reversible Variable Length Codes), EREC (Error Resilience Entropy Coding), MDC (Multiple-Description Coding), RFS (Reference Frame Selection), AIR (Adaptive Intra Refresh), etc. [3]. On the other hand, FEC (Forward Error Correction) and ARQ (Automatic Retransmission reQuest) are the two major schemes for channel protection. For error-resilient source coding, AIR is the most commonly used tool among the existing methods [1,16-22], because it does not need to make any change for standard video decoders, which is important in terms of cost and convenience for many practical applications. The major issue for AIR is to decide which

MBs need to be intra-refreshed. If feedback information is available, the error tracking methods proposed in [1,16,17] can precisely locate the corrupted MBs to perform intra-refresh so that the error propagation can be terminated effectively and quickly. Such methods, however, require extra cost of computation and memory for tracking the locations of erroneous blocks. The cost grows significantly as the channel round-trip delay increases. Several other methods have been proposed to estimate the error propagation effect due to packet loss [18-21], but some of these methods cannot dynamically adapt to the varying channel conditions or involve complicated computation for estimating the amount of drifting error due to packet loss. The CBERC (content-based error-resilient coding) scheme proposed in [22] takes video content into account in making coding mode decisions by using the concealment error to identify GOBs (Group of Blocks) of more importance. However, using error concealment error only without considering motion information may not be able to capture the error propagation effect very well.

There have been a few research works about error resilient video transcoding [10-14]. For example, an error-resilient MPEG-2 transcoding scheme based on EREC is proposed in [10]. In this method, the incoming bitstream is reordered without adding redundancy such that longer VLC (Variable Length Coding) blocks fill up the spaces left by shorter blocks in a number of VLC blocks which form a fixed-length EREC frame. Such fixed-length EREC frames of VLC codes are then used as synchronization units, where only one EREC frame, rather than all the codes between two synchronization markers, will be dropped should any VLC code in the EREC frame be corrupted due to transmission error. The method proposed in [11] suggests a rate-distortion framework with analytical models that characterize the error propagation of corrupted video bitstream subjected to bit errors. These models are then used to guide the use of spatial and temporal localization tools: synchronization marker and intra-refresh so as to obtain the optimal combinations of spatial error-resilience, temporal error-resilience, and transmission bitrate. This method can achieve good performance, however, its computational complexity may be too high to be used for real-time applications. The work in [12] proposes an error-resilient transcoder for GPRS (general packet ratio services) mobile-access networks. The transcoding process is performed at a video proxy located at the edge of two or more networks. Two error-resilience tools: the AIR and RFS methods with FCS (Feedback Control Signaling), are exploited adaptively to reduce error effects, while preserving the transmission rate management feature of the video transcoders. In [13], a multiple-description FEC (MD-FEC) based transcoding scheme is proposed. This method adopts the (N, i, $N - i + 1$) Reed-Solomon erasure-correction block code to protect the ith layer of an N-layer scalable video. A special multiple-description packetization scheme is presented so as to ensure the ith layer to be decodable when i or more descriptions are received by the decoder. The method in [14] proposes to implement an ARQ proxy at the base station of a wireless communication system for handling ARQ requests and tracking errors so as to reduce retransmission delays as well as enhance the error resilience. The ARQ proxy selectively resends important lost packets (e.g., packets with header information and motion vectors) detected through the retransmission requests from client terminals, while dropping the remaining non-important packets (packets carrying DCT coefficients) for rate shaping. A transcoder is used to compensate for the mismatch error between the front-end video encoder and the client decoders caused by the dropped packets.

In this work, we propose a novel two-pass transcoding scheme for inserting error-resilience features to a compressed video at the media gateway of a three-tier streaming system. AIR is adopted in our proposed transcoder as the error-resilience coding tool so that standard video decoders can be used in the client terminals. The proposed transcoder can adaptively adjust the intra-refresh rate according to the video content and the channel's packet-loss rate to protect MBs of high loss-impact values against packet loss. In the first-pass encoding, the encoder estimates the amount of error propagation to each MB when packet loss occurs, and then generates side information as transcoding hints for use at the transcoder. In the second-pass transcoding, the error-resilient transcoder adaptively determines the intra-refresh rate and the locations of MBs to perform intra-refresh according to the side information.

The rest of this paper is organized as follows. Section 2 presents the proposed two-pass error-resilient transcoder. Detailed operations in each pass are elaborated. Section 3 reports the experimental results of the proposed algorithms and the comparison with other commonly used methods. The run-time complexity and extra overhead cost are also analyzed. Finally, conclusions are drawn in Section 4.

2. Proposed Two-Pass Intra-Refresh Transcoding Scheme

Fig. 3 shows the proposed two-pass error-resilient transcoder architecture for a three-tier video streaming system. At the first-pass front-end encoding, in addition to the standard encoding process, the encoder also utilizes the motion vectors generated in the encoding process and the estimated concealment distortion to estimate the error-propagation effect at the MB and frame levels within a GOP. The MBs are then ranked by the estimated amount of error propagation. As a result, the MB-level rank-order information and the frame-level error-propagation estimates are stored in the streaming server as the side information. This side information is sent to the intermediate transcoder as transcoding hints to guide the error-resilient transcoding operation while streaming the video to client terminals.

In the second-pass transcoding process, the transcoder uses the side information received from the streaming server and the channel statistics (e.g., the packet-loss rate) collected from the feedback channel to determine the intra-refresh allocation for each frame of a GOP. The transcoder then performs intra-fresh on a number of high-priority MBs with highest loss-impact factors based on the intra-refresh allocation. The key idea behind the proposed transcoding scheme is to stop the error propagation in the current frame by performing intra-refresh on those MBs which reference to a high loss-impact prediction block of the pervious frame, thereby having a high possibility of being corrupted.

In the proposed scheme, most of the computation is done in the first-pass front-end encoding, which usually does not need to be done in real-time for prestored video applications. Only a small amount of computation is left to the second-pass transcoding, which usually has to meet the real-time requirement. In the first-pass encoding, the major computation is to analyze the error propagation effect using motion information and concealment error. The computational complexity for error-propagation estimation is relatively high, but usually can be done off-line.

2.1. Loss-impact estimation in the first-pass encoding

Due to the inter-frame coding techniques used in video compression, drifting error caused by lost MBs will propagate to subsequent frames until reaching an intra-refresh point (e.g., an I-frame, or an intra-coded block). To capture such error propagation effect, we first define the pixel-level loss-impact (LI) metric as the product of two parameters: PRC (Pixel Reference Count) and PCE (Pixel Concealment Error), to characterize the amount of pixel-wise error propagation as follows.

$$LI(x,y,n) = PCE(x,y,n) \times PRC(x,y,n) \tag{1}$$

where $PRC(x,y,n)$ represents the frequency of pixel (x,y) of frame n being referenced by pixels in the following frames within a GOP in the motion-compensated prediction process as illustrated in Fig. 4. $PRC(x,y,n)$ can be calculated recursively by summing up the individual reference counts of pixels in frame n+1 which reference to pixel (x,y) of frame n in the reverse tracking order from the last frame to the first frame of a GOP as in (2).

$$PRC(x,y,n) = \begin{cases} \displaystyle\sum_{(x',y',n+1) \text{ points to } (x,y,n)} PRC(x',y',n+1) & 1 \le n < N_{\text{GOP}} \\ 1 & n = N_{\text{GOP}} \end{cases} \tag{2}$$

where $PCE(x,y,n)$ denotes the norm of concealment error of pixel (x,y) of frame n should this pixel be corrupted. In this work, the zero-motion error concealment scheme [24] is adopted. Therefore we obtain

$$PCE(x,y,n) = \left| f(x,y,n) - f(x,y,n-1) \right|^2 \tag{3}$$

where $f(x,y,n)$ is the pixel value of pixel (x,y) in frame n.

After estimating the pixel-level loss-impact values, as depicted in Fig. 5, we use the motion information to map pixel-level loss-impact values of the pervious frame to obtain the current-frame's MB-level error-propagation (from the previous frames) as follows.

$$EP_{\text{MB}}(m,n) = \sum_{(x,y)\in \text{MB}_m} LI(x+\text{MV}_x, y+\text{MV}_y, n-1) \tag{4}$$

where m is the MB index in a frame; (x,y) is the pixel coordinate; n is the time index; $(\text{MV}_x,\text{MV}_y)$ is the associated motion vector of pixel (x,y). A high EP_{MB} value implies that the MB references to a prediction block with high concealment error and/or a high accumulated reference count. Therefore performing intra-refresh on this MB usually can more effectively terminate the error propagation due to packet loss in the previous frames than on MBs with lower EP_{MB}.

Finally, all EP_{MB}'s in each frame are summed up to estimate the frame-level error-propagation as follows.

$$EP_n = \sum_{m=1}^{N_{\text{MB}}^{\text{F}}} EP_{\text{MB}}(m,n) \tag{5}$$

where N_{MB}^{F} is the number of MBs in a frame.

After obtaining the above features in the first-pass front-end encoding, two kinds of information are extracted and stored at the streaming server which will be sent to the intermediate transcoder as side information to enhance error resilience while streaming. First, all MBs in a frame are sorted by the EP_{MB} values, and the ranks of EP_{MB}'s of MBs in one frame are stored as side information at the streaming server. Second, the frame-level error propagation estimate of each frame defined in (5) is also stored in the server.

2.2. Content-aware transcoding using prioritized intra-refresh

In the second-pass transcoding, we propose a prioritized intra-refresh scheme to determine the intra-refresh rate and the intra-MB allocation strategy for each GOP so as to adapt the transcoded video to varying network conditions. The proposed transcoding method is divided into three levels. The first-level process determines the total number of MBs to be intra-coded within a GOP according to the estimated channel packet loss rate (*PLR*), the average frame-level error propagation value in a GOP, and a control parameter as described as follows.

$$N_{\text{intra}}^{\text{GOP}} = \frac{\frac{1}{N_{\text{GOP}}} \sum_{n=1}^{N_{\text{GOP}}} EP_n \times PLR}{TH_{\text{intra}}} \tag{6}$$

where $N_{\text{intra}}^{\text{GOP}}$ is the total number of MBs to be intra-refreshed in a GOP (i.e., GOP-level intra-refresh allocation); N_{GOP} is the GOP size; *PLR* is the channel packet loss rate which can be estimated via the client feedback information (e.g., the RTCP reports [25]); TH_{intra} is a scaling parameter.

In (6), the GOP-level intra-refresh allocation, $N_{\text{intra}}^{\text{GOP}}$, is proportional to the product of the estimated amount of error-propagation and the estimated channel packet-loss rate. According to (6), the higher the estimated error-propagation factor and/or the packet-loss rate, the more the intra-refresh allocation. The channel packet-loss rate *PLR* is updated every GOP to capture frequently changing network conditions. The scaling parameter TH_{intra}, which is determined empirically in our work, is used to characterize the relationship between $N_{\text{intra}}^{\text{GOP}}$ and the error-propagation effect in a GOP.

After determining the GOP-level intra-refresh budget, one intuitive approach to performing the next-level prioritized intra-refresh is to sort all MBs in a GOP by EP_{MB} and select the $N_{\text{intra}}^{\text{GOP}}$ top-ranked MBs to perform intra-refresh. This can protect the MBs with highest possibility of being corrupted by error propagation in a GOP, but may lead to two drawbacks. First, such method usually makes most intra-refreshed MBs concentrated in the first half frames of the GOP, since they are usually of higher loss-impact values, thereby cannot effectively mitigate the drifting error caused by lost blocks belonging to the later frames of the GOP. Second, should there be too many intra-refreshed MBs crowded together in a frame, the rate-control scheme has to increase the quantization step-size to avoid buffer overflow, leading to significantly lower visual quality.

To address this problem, we propose a three-level intra-refresh allocation scheme. In the proposed scheme, after determining the GOP-level intra-refresh budget, rather than distributing the budget to the top-ranked MBs directly, the next step is to distribute the intra-refresh budget into all frames in a GOP intermediately. In order to determine how many MBs need to be intra-refreshed in a frame, the following frame-level intra-refresh distribution method is used.

if $n = 2$ (i.e., the first P-frame in a GOP)

$$N_{\text{intra}}^{n} = \frac{EP_n}{\sum_{i=n}^{N_{\text{GOP}}} EP_i} \times N_{\text{intra}}^{\text{GOP}} \tag{7}$$

else if $3 \le n \le N_{\text{GOP}}$

$$N_{\text{intra}}^{n} = \frac{EP_n}{\sum_{i=n}^{N_{\text{GOP}}} EP_i} \times \left(N_{\text{intra}}^{\text{GOP}} - \sum_{i=2}^{n-1} N_{\text{intra}}^{i} \right) \tag{8}$$

endif

if $N_{\text{intra}}^{n} > k_{\text{MB}} N_{\text{MB}}^{\text{F}}$ **then** $N_{\text{intra}}^{n} = k_{\text{MB}} N_{\text{MB}}^{\text{F}}$

where n is the frame index in a GOP; N_{intra}^{n} is the number of MBs to be intra-coded in frame n; N_{MB}^{F} is the number of MBs in a frame; k_{MB} ($0 \le k_{\text{MB}} \le 1$) is a control parameter to constrain the number of intra-coded blocks in a frame not to exceed an upper limit (say, $k_{\text{MB}} N_{\text{MB}}^{\text{F}}$). In this work, we simply set $k_{\text{MB}} = 1$.

In (7) and (8), the frame-level intra-refresh distribution is made for the nth frame in a GOP based on the ratio of the frame's loss-impact factor with respect to the sum of the loss-impact factors of all the frames from the nth frame to the end of the GOP. As a result, for the nth frame of a GOP, the final MB-level intra-refresh allocation selects a total of N_{intra}^{n} MBs with top-ranked EP_{MB} values to perform intra-refresh according to the ranking information prestored in the streaming server. Such three-level intra-refresh allocation algorithm makes intra-refreshed MBs distributed among all frames in a GOP so as to avoid the above-mentioned drawbacks caused by directly distributing the GOP-level budget to the top-ranked MBs of a GOP without using the proposed intermediate frame-level distribution scheme.

3. Experimental Results

Four 300-frame CIF (352×288) test sequences as listed in Table 1 are used in our experiments. These sequences are pre-encoded using an MPEG-4 public-domain software encoder at 30 fps and 384 Kbps with the I-B-P GOP (Group of Pictures) structure with (N_{GOP},M) = (30,2), where N_{GOP} is the GOP size, and M is the distance between two anchor I/P-frames (i.e., M-1 B-frames are inserted between two anchor frames). We implemented a cascaded pixel-domain transcoder [6] based on the MPEG-4 public-domain software to perform the intra-refresh transcoding. The output bit-rate, after inserting intra-refresh MBs, is regulated to the same bit-rate of the input video (i.e., 384 kbps) by using the MPEG-4 TM-5 rate-control scheme. For video transmission, a slice which contains one row of MBs is encapsulated into one packet. In this work, we use a two-state Markov model to simulate the channel conditions. We adopt a simplified Gilbert channel at the packet level [23] to generate the packet-loss patterns with four packet loss rates (*PLR*): 5%, 10%, 15%, and 20%, respectively. The average burst length is set to 1 to simulate the random packet-loss situations. For slow-fading channels which will result in longer burst losses, packet interleaving techniques can be used to spread a burst loss into individual single-packet losses to facilitate the error control process if the introduced complexity and delay are acceptable [3].

For performance evaluation, three other intra-refresh methods: random intra-refresh [21], regular intra-refresh [21] and CBERC [22] are also implemented and compared with the proposed method. Suppose the average number of intra-refreshed MBs in a frame is m. In the random intra-refresh scheme, m MBs are randomly selected to be intra-refreshed for each frame. By regular intra-refresh, the MB positions for intra-refresh are 1~m in the first frame, m+1~2m in the second frame, and so on. If all positions have been refreshed once, the refresh pattern is repeated again.

In the proposed method, the scaling factor of intra-refresh rate, TH_{intra}, in Eq. (6) is determined empirically. Fig. 6 shows the frame-by-frame PSNR with different TH_{intra} values for three test sequences when PLR = 10%. We can observe that TH_{intra} = 1200 stably achieves the best performance for every sequence. Therefore we adopt TH_{intra} = 1200 for all the sequences at different packet-loss rates.

The average PSNR performance comparison with different methods and channel conditions is shown in Table 1 and Fig. 7. Figs. 8-11 depict the frame-by-frame PSNR performance comparisons of the proposed method with the other three intra-refresh methods for the four test sequences. Since I-frames are important in terminating the error propagation, a corrupted I-frame may cause serious error propagation. Furthermore, MBs closer to the I-frame in a GOP are usually more important than other farther MBs. This is because, if these important MBs are corrupted, the resultant error propagation will be more serious, since the drifting error will propagate to more frames until reaching the next I-frame. This kind of effects can be observed in frames #91-#151 in Fig. 8(a), where large PSNR drops can be observed because many of MBs of higher importance are corrupted. In this case, the proposed method can mitigate the error propagation caused by the loss of important packets (e.g., packets close to I-frames) more effectively than the other methods. Moreover, the proposed method usually can also mitigate serious error propagation at a faster speed. Similar situations can also be observed in other figures. Figs. 12-15 illustrate some reconstructed frames using different transcoding methods for subjective performance comparison. In most cases, the proposed method outperforms the other three intra-refresh schemes at different packet loss rates. For the "Dance" sequence, the proposed method can achieve up to 1.6 dB improvement over the CBERC method.

Table 1. Average PSNR comparison between different intra-refresh schemes for four test sequences (in dB) for different packet loss rates (average burst length = 1)

Sequence	Method	Packet-Loss Rate (*PLR*)			
		5%	10%	15%	20%
Foreman	Error Free	35.80			
	Non-E.R.	29.73	26.63	25.37	22.95
	Proposed	**31.51**	**30.20**	**29.21**	**28.46**
	Regular	30.30	28.88	27.84	26.99
	Random	30.29	28.78	27.73	26.85
	CBERC	31.14	30.12	29.20	28.47
Coastguard	Error Free	33.53			
	Non-E.R.	29.46	27.34	26.23	24.55
	Proposed	**30.06**	**29.07**	**28.08**	**27.51**
	Regular	29.92	28.86	28.02	27.25
	Random	29.88	28.84	27.97	27.15
	CBERC	29.79	28.67	28.07	27.24
Dancer	Error Free	39.71			
	Non-E.R.	30.87	28.25	26.51	24.20
	Proposed	**32.09**	**31.3**	**30.29**	**29.12**

	Regular	32.00	30.56	29.33	28.08
	Random	31.94	30.62	29.09	28.02
	CBERC	31.23	29.79	28.60	28.11
Salesman	Error Free	39.81			
	Non-E.R.	37.74	35.76	34.56	32.98
	Proposed	**37.85**	**36.94**	**36.15**	**35.14**
	Regular	37.61	36.28	34.89	33.92
	Random	37.54	36.27	34.79	33.82
	CBERC	37.79	36.74	35.84	35.10

In order to evaluate the individual gains from the frame-level and MB-level intra-refresh distribution schemes, we also performed two experiments. The first is to combine the proposed frame-level distribution scheme with three MB-level intra-refresh strategies: random, regular, and CBERC. The second is to use the uniform distribution at the frame level, than apply the proposed MB-level intra-refresh scheme. Fig. 16 show the results of the experiments, where "Frame+CBERC," "Frame+Random," and "Frame+Regular" represent the combinations of the proposed frame-level distribution scheme with the CBERC, random, and regular MB-level intra-refresh strategies, respectively. "Uniform+MB" represents the combination of uniform frame-level distribution with the proposed MB -level distribution. We can observe that applying the proposed frame-level distribution to the random, regular, and CBERC MB-level distributions only provides similar performances compared to those which adopt the uniform frame-level distribution. The reason is, although the proposed frame-level distribution scheme will allocate more intra-MB budgets to those frames of relatively higher loss-impact (typically the first-half frames), the random and regular distributions may not place these intra-MBs to appropriate locations. If the intra-refresh budgets are not used appropriately, the relatively less intra-MB allocations in the later frames may not be able to terminate the error propagation effectively. The CBERC distribution intra-refreshes those MBs with concealment error larger than a threshold without setting frame-level budgets, so the proposed frame-level distribution only provides comparable performance. The error-resilience gain achieved by the combination of the uniform frame-level distribution and the proposed MB-level distribution is also not very stable. The reason is, although the proposed MB-level distribution scheme tends to protect MBs of highest loss impacts, the uniform frame-level distribution may not provide enough budget to intra-refresh all the high loss-impact MBs entirely. The combination of the proposed frame-level and MB-level distribution schemes can achieve the best performance stably.

Table 2 shows the run-time analysis of the first-pass encoding and second-pass transcoding on an Intel Pentium-III 1-GHz PC. The proposed encoder and transcoder are implemented based on the MPEG-4 public domain software. With the proposed error-propagation estimation method, the first-pass encoding consumes significantly more time than the original one. On the other hand, the proposed method does not increase the computational complexity of second-pass transcoding. Actually sometimes it consumes even less time than the original transcoder for two reasons. First, the computation for intra-refresh decision in (5)-(7) in the second-pass transcoding is almost negligible compared to the whole transcoding process. Second, the error-resilient transcoding will increase the number of intra-coded MBs, thereby reducing the computation since the computational cost for intra-coding is much lower than that for inter-coding. Such asymmetric computation requirement is suitable for the three-tier prestored video streaming application scenario.

Table 2. Run-time analysis of the first-pass encoding and second-pass transcoding

Sequence	Encoding Time		Transcoding Time	
	original	proposed	Non-error resilient	Error-resilient
Foreman	11.0 s	23.7 s	15.1 s	14.5 s
Coastguard	11.1 s	23.7 s	14.9 s	14.7 s
Dancer	11.0 s	23.7 s	12.2 s	12.2 s
Salesman	11.2 s	23.7 s	15.2 s	15.0 s

In the applications of two-tier video streaming where no intermediate transcoder is placed between the server and clients, the transcoding side information is stored in and used for the server only, leading to extra storage cost, but will not incur extra communication cost. On the other hand, under the three-tier streaming scenario, the proposed method has to send the side information from the streaming server to the intermediate transcoder, which will consume extra bandwidth. For example, the maximum number of rank-orders of impact values is 99 for a QCIF (176×144) frame, and 396 for a CIF frame, leading to 7 and 9 bits per MB to represent the full rank-orders of impact values, respectively. The resultant overhead cost is about 5.29% for a QCIF video coded at 384kbps and 30 fps. One method of reducing the overhead cost is to classify all MBs in a frame into fewer (say, 2^m classes with $m < 7$) rank-orders of impact values, rather than the full rank-orders of MBs. According to our experiments, if eight classes (i.e., 3 bits/MB) are used for ranking the MBs in first-half frames and four classes for ranking MBs in last-half frames, the overhead cost is about 1.9% and the quality degradation due to such simplification is only 0.04 dB, which is almost negligible. To further reduce the overhead cost, the ranking and intra-refresh can be performed on a group of two or more consecutive MBs as a unit, rather than on individual MBs.

4. Concluding Remarks

In this paper, we proposed a novel two-pass error-resilient transcoding scheme by using prioritized intra-refresh. The first-pass encoding process of the proposed method estimates the pixel-wise loss-impact using an error-tracking technique. The estimated pixel-wise mismatch error is subsequently used to estimate the amount of error propagation impact from previous frames for each MB. After sorting the error propagation factors of all MBs in a frame, the rank of error-propagation effect of each MB and the frame-level error-propagation impact values are stored in the streaming server as side information for the future transcoding.

In the second-pass transcoding, the transcoder first determines the GOP-layer intra-refresh budget according to the side information and estimated channel condition, and subsequently uses the side information to distribute the intra-refresh budget into each frame of a GOP. The extra computational complexity required for this computation is almost negligible, thereby making it suitable for real-time transcoding applications.

The proposed algorithm can effectively mitigate the error propagation due to packet-loss and improve the quality significantly. The degree of error resilience can be dynamically adjusted to adapt to a channel with time-varying error characteristics which can be estimated using the statistics collected through the feedback channel.

References

[1] B. Girod, N. Färber, Feedback-based error control for mobile video transmission, Proc. IEEE 87 (10) (1999) 1707-1723.

[2] Z. He, J. Cai, C.-W. Chen, Joint source channel rate-distortion analysis for adaptive mode selection and rate control in wireless video coding, IEEE Trans. Circuits Syst. Video Technol. 12 (6) (2002) 511-523.

[3] Y. Wang, Q.-F. Zhu, Error control and concealment for video communication: A review, Proc. IEEE 86 (5) (1998) 974–997.

[4] T. Warabino, S. Ota, D. Morikawa, M. Ohashi, H. Nakamura, H. Lwashita, F. Watanane, Video transcoding proxy for 3G wireless mobile internet access, IEEE Trans. Communication Magazine 38 (2000) 66-71.

[5] H. Sun, W. Kwok, J. Zdepski, Architectures for MPEG compressed bitstream scaling, IEEE Trans. Circuits Systems Video Technol. 6 (1996) 191-199.

[6] J. Youn, M.-T. Sun, C.-W. Lin, Adaptive motion vector refinement for high performance transcoding, IEEE Trans. Multimedia 1 (1) (1999) 30-40.

[7] P. A. A. Assunção, M. Ghanbari, A frequency domain video transcoder for dynamic bit rate reduction of MPEG-2 bit streams, IEEE Trans. Circuits Systems Video Technol. 8 (8) (1998) 953-967.

[8] T. Shanableh, M. Ghanbari, Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats, IEEE Trans. Multimedia 2 (2) (2000) 101-110.

[9] A. Vetro, C. Christopoulos, H. Sun, Video transcoding architectures and techniques: an overview, IEEE Signal Processing Magazine 20 (3) (2003) 18-29.

[10] R. Swann, N. Kingsbury, Transcoding of MPEG-II for enhanced resilience to transmission errors, in: Proceedings of the IEEE International Conference on Image Processing, vol. 2, 1996, pp. 813-816.

[11] G. de los Reyes, A. R. Reibman, S. F. Chang, J. C. I. Chuang, Error-resilient transcoding for video over wireless channels, IEEE Journal on Selected Areas in Communications 18 (6) (2000) 1063-1074.

[12] S. Dogan, A. Cellatoglu, M. Uyguroglu, A. H. Sadka, A. M. Kondoz, A. M. Kondoz, Error-resilient video transcoding for robust internetwork communications using GPRS, IEEE Trans. Circuits Systems Video Technol. 12 (6) (2002) 453-464.

[13] R. Puri, K.-W. Lee, K. Ramchandran, V. Bhargavan, An integrated source transcoding and congestion control paradigm for video streaming in the Internet, IEEE Trans. Multimedia 3 (1) (2001) 18-32.

[14] T.-C. Wang, H.-C. Fang, L.-G. Chen, Low delay and error robust wireless video transmission for video communication, IEEE Trans. Circuits Systems Video Technol. 12 (12) (2002) 1049-1058.

[15] A. Vetro, J. Cai, C.-W. Chen, Rate-reduction transcoding design for wireless video streaming, Journal of Wireless Communications and Mobile Computing 2 (6) (2002) 549-552.

[16] P. C. Chang, T. H. Lee, Precise and fast error tracking for error-resilient transmission of H.263 video, IEEE Trans. Circuits Systems Video Technol. 10 (6) (2000) 600-607.

[17] E. Steinbach, N. Färber, B. Girod, Standard compatible extension of H.263 for robust video transmission in mobile environments, IEEE Trans. Circuits Systems Video Technol. 7 (6) (1997) 872-881.

[18] G. Cote, S. Shirani, F. Kossentini, Optimal mode selection and synchronization for robust video communications over error-prone networks, IEEE Journal on Selected Areas in Communications 18 (6) (2000) 952-965.

[19] R. Zhang, S. L. Regunathan, K. Rose, Video coding with optimal inter/intra-mode switching for packet loss resilience, IEEE J. Select. Areas Communications 18 (6) (2000) 966-976.

[20] R. Zhang, S. Regunathan, K. Rose, Optimized video streaming over lossy networks with real-time estimation of end-to-end distortion, in: Proceedings of the IEEE International Conference on Multimedia and Expo, vol. 1, 2002, pp. 861-864.

[21] G. Cote, F. Kossentini, Optimal intra coding of blocks for robust video communication over the internet, Signal Processing: Image Communication 15 (1999) 25-34.

[22] W.-H. J. Chen, J.-N. Hwang, The CBERC: A content-based error-resilient coding technique for packet video communications, IEEE Trans. Circuits Systems Video Technol. 11 (8) (2001) 974-980.

[23] E. N. Gilbert, Capacity of a burst-noisy channel, Bell Syst. Tech. J., Vol. 39, 1960, pp. 1253-1265.

[24] W. M. Lam, A. R. Reibman, B. Liu, Recovery of lost or erroneously received motion vectors, in: Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing, vol. 5, 1993, pp. 417-420.

[25] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, RTP: a transport protocol for real-time applications, Internet Engineering task Force, *RFC 1889*, Jan. 1996.

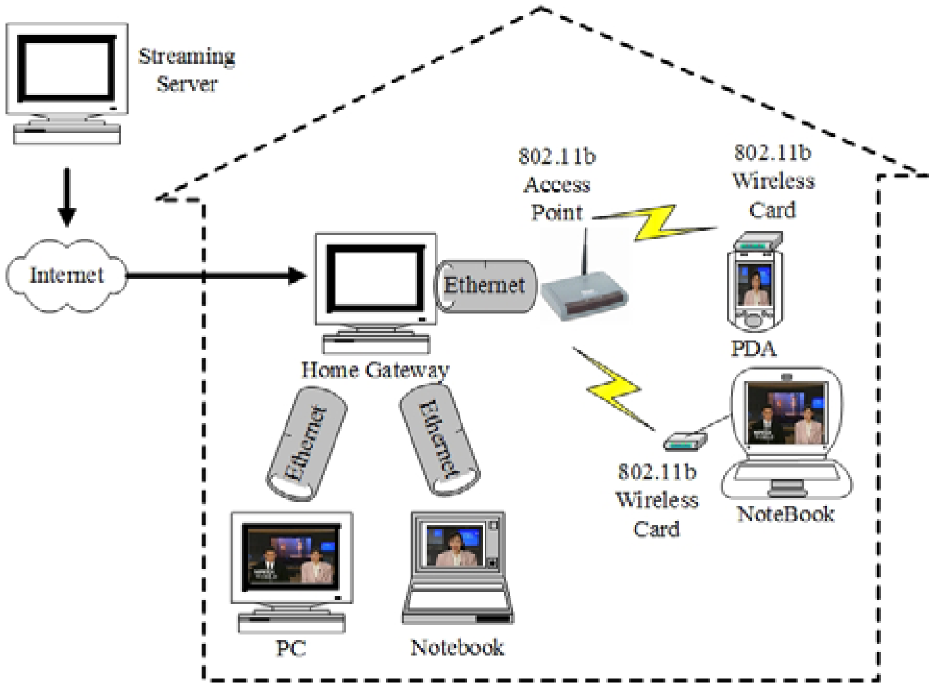


Fig. 1. An example of three-tier video streaming system for home networking.

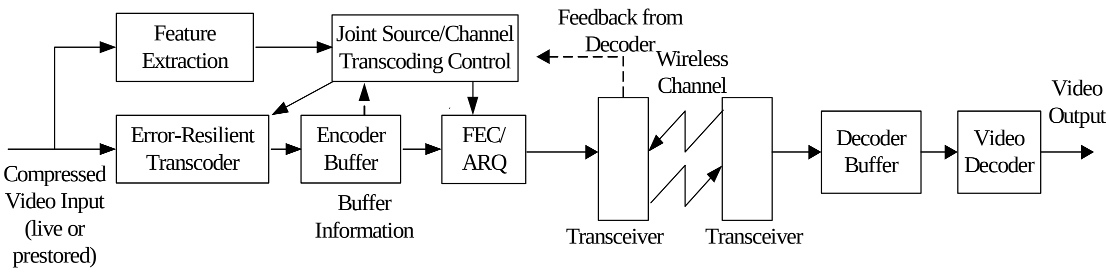
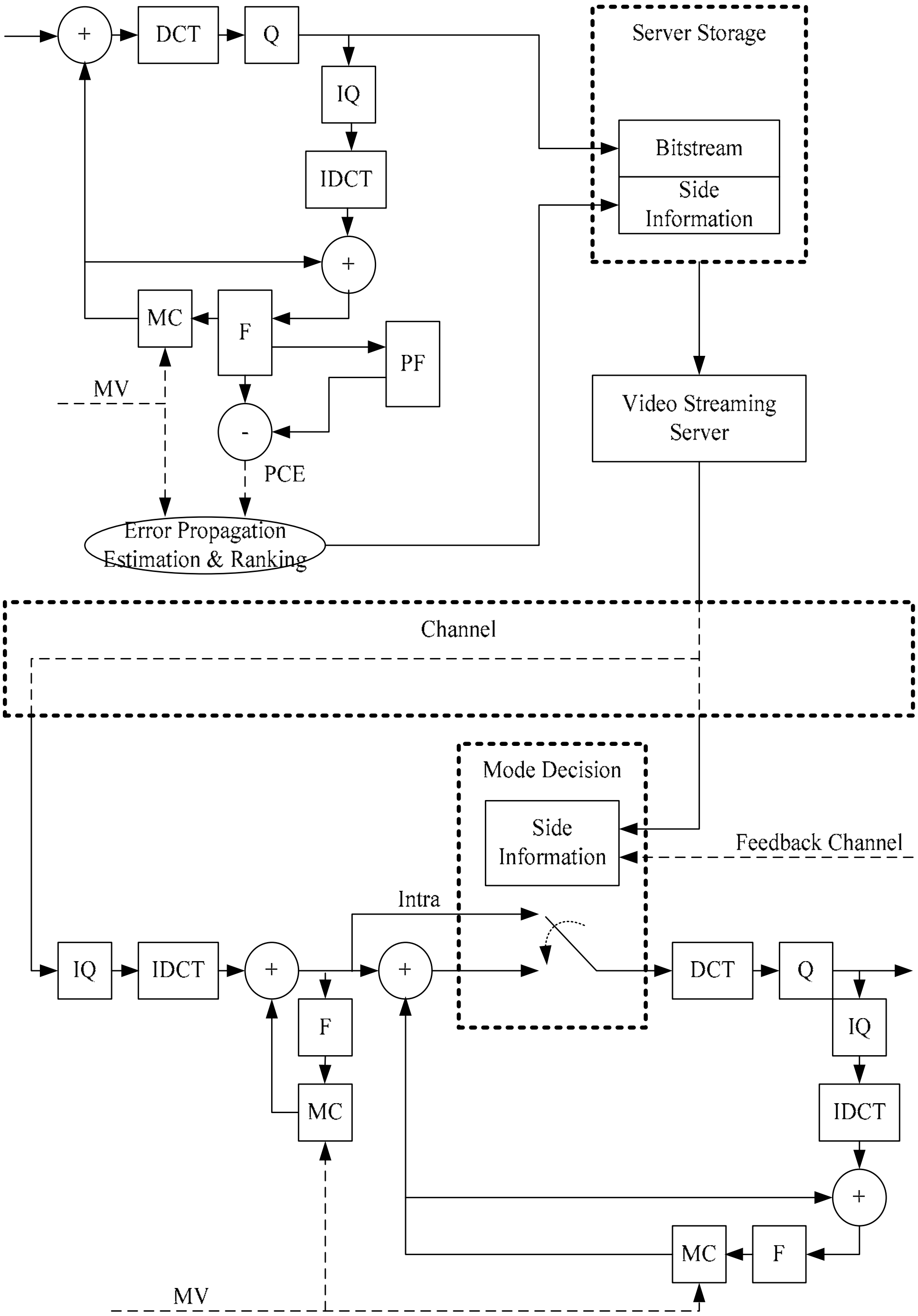


Fig. 2. A typical system framework of error-resilient video transcoder.



DCT：Discrete Cosine Transform
IDCT：Inverse Discrete Cosine Transform
Q：Quantization
IQ：Inverse Quantization

MC：Motion Compensation
MV：Motion Vector
F：Frame Memory
PF：Pervious Frame Memory
PCE：Pixel Concealment Error

Fig. 3. Proposed architecture of two-pass error-resilient transcoder.

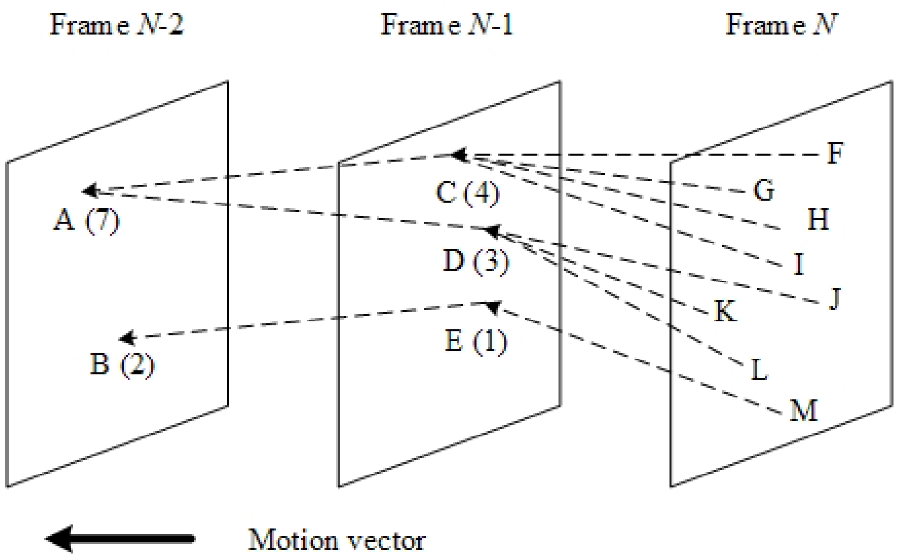


Fig. 4. An illustration of calculating the pixel reference count (PRC). Assume frame *N* is the last frame of a GOP, the number in the braces indicate the PRC of a pixel.

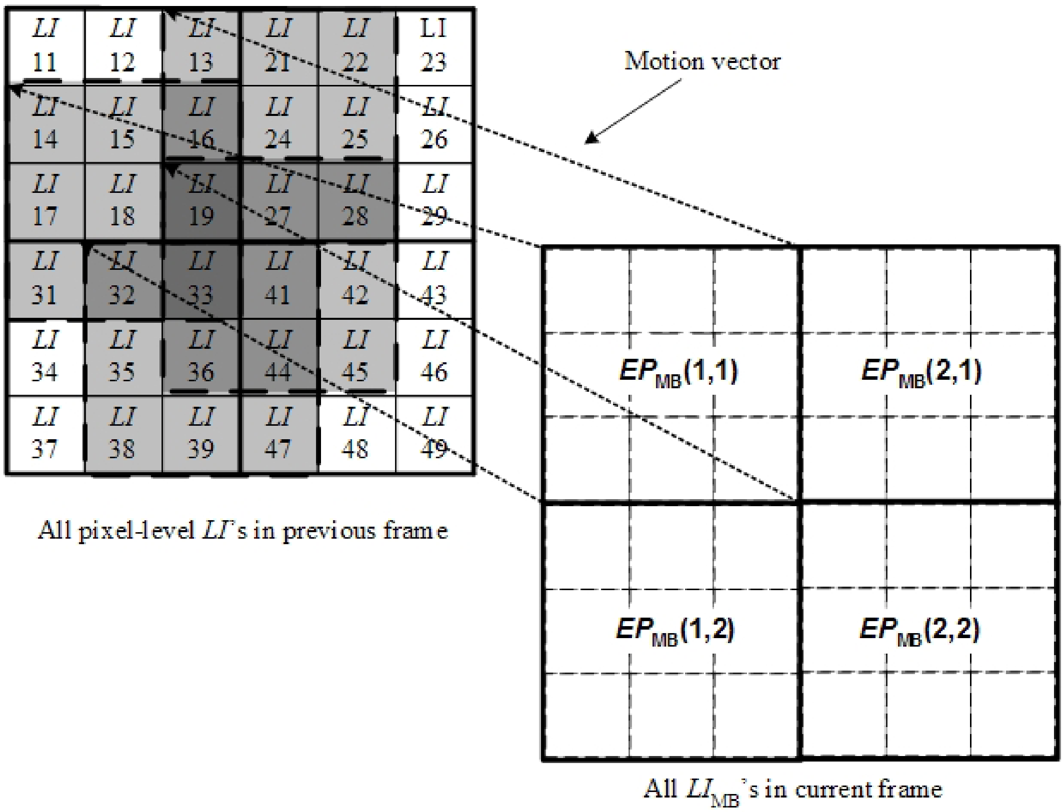


Fig. 5. Illustration of using motion vectors to map pixel-level loss-impact values from the previous frame to obtain MB-level error-propagation impact values in the current frame.

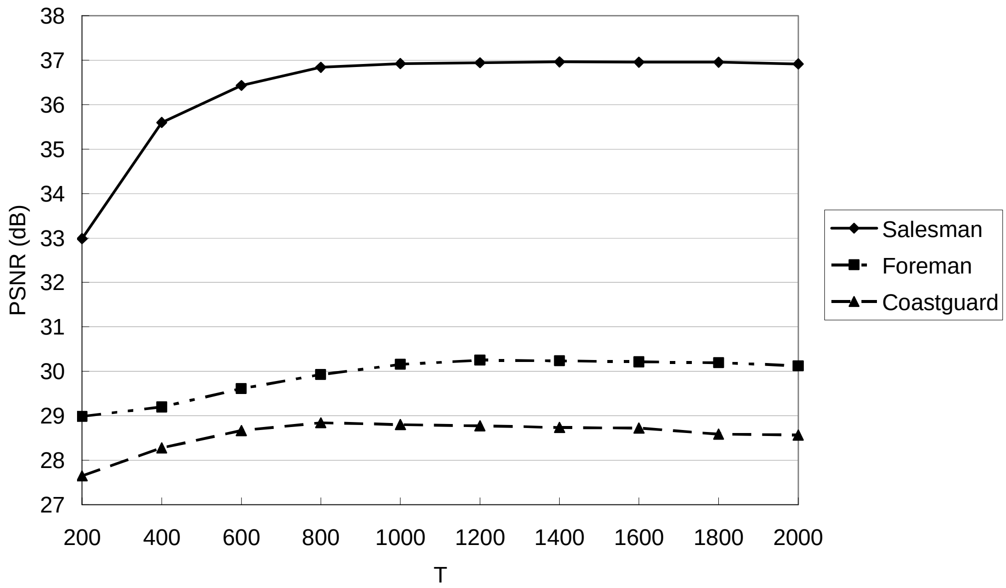
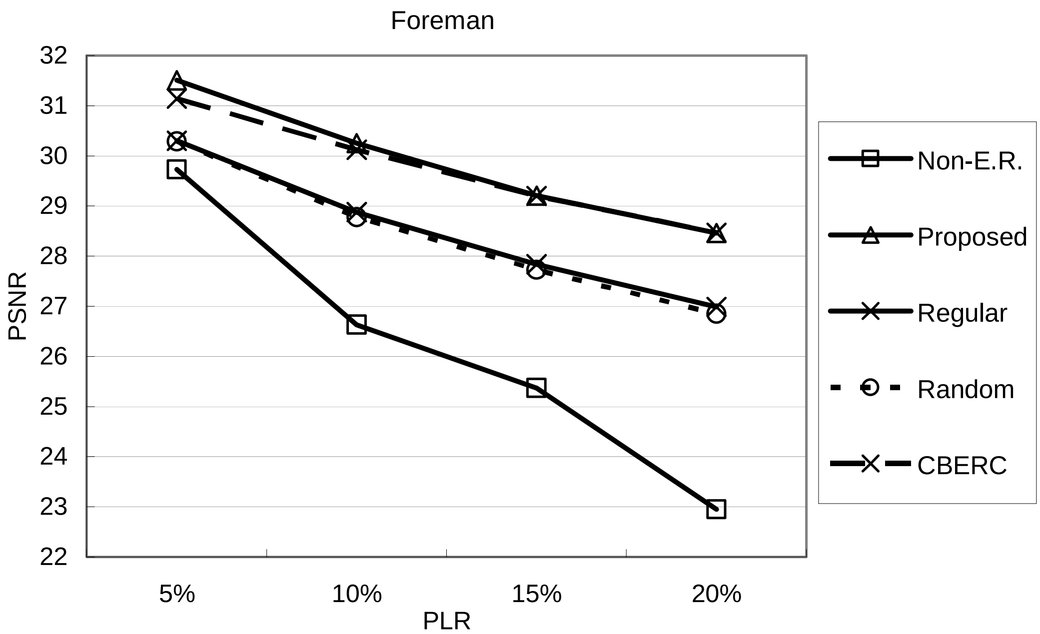
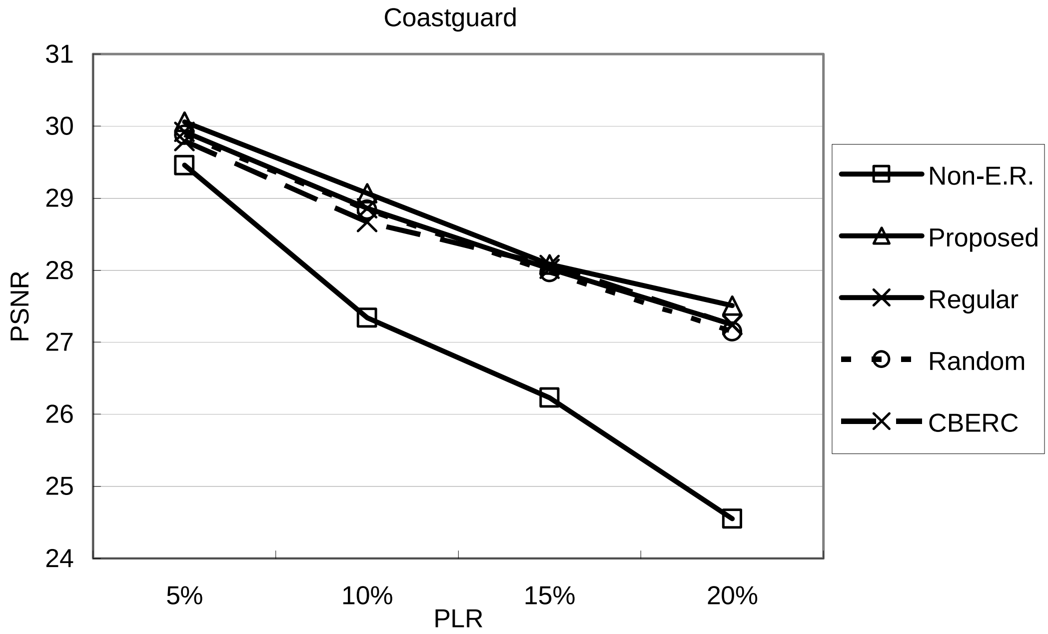


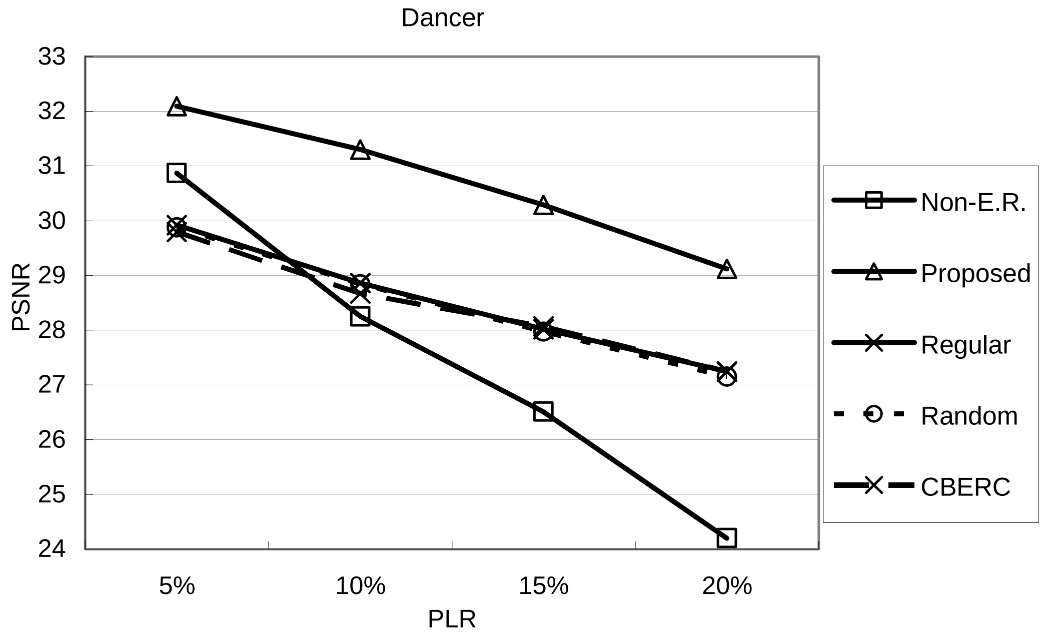
Fig. 6. Effect of changing the intra-refresh parameter (*PLR* = 10%).



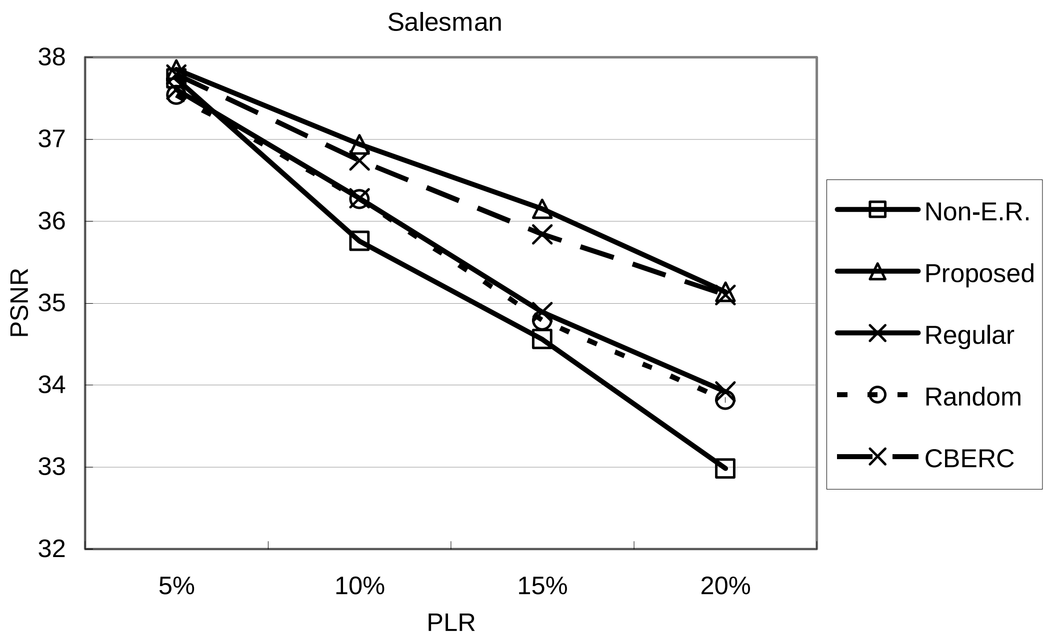
(a) Foreman



(b) Coastguard

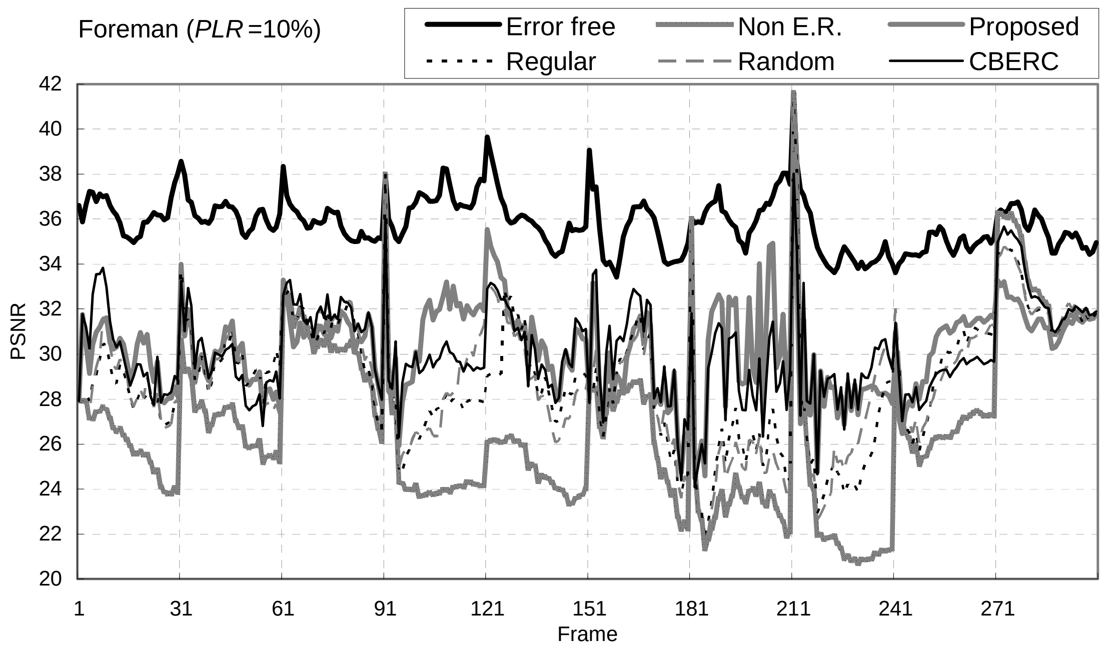


(c) Dancer

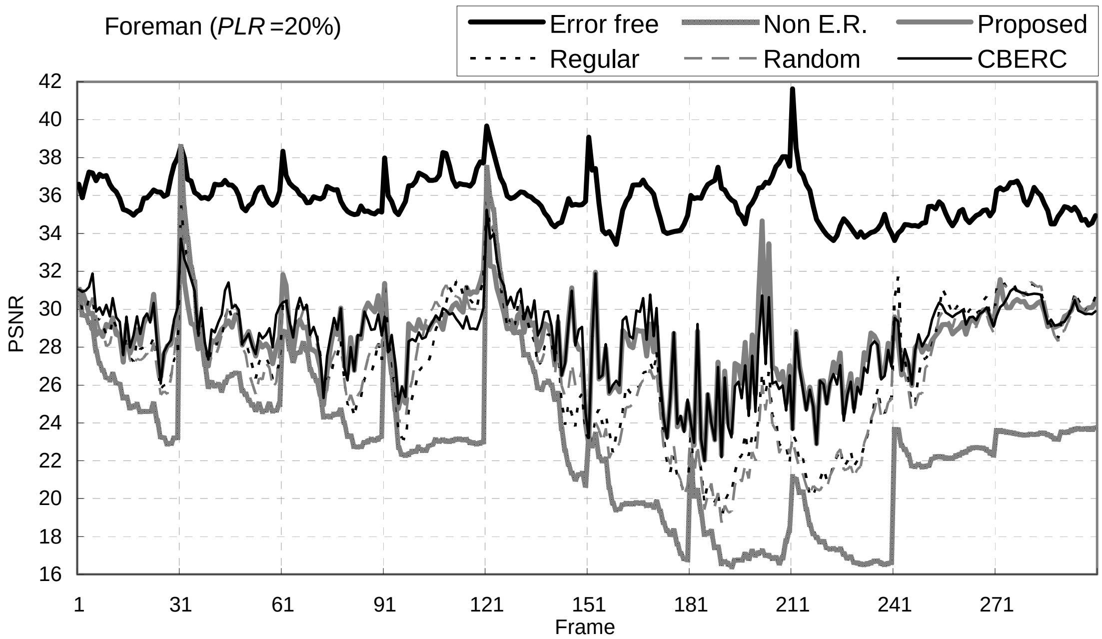


(d) Salesman

Fig. 7. Average PNSR performance comparison using four intra-refresh methods under four different channel conditions.

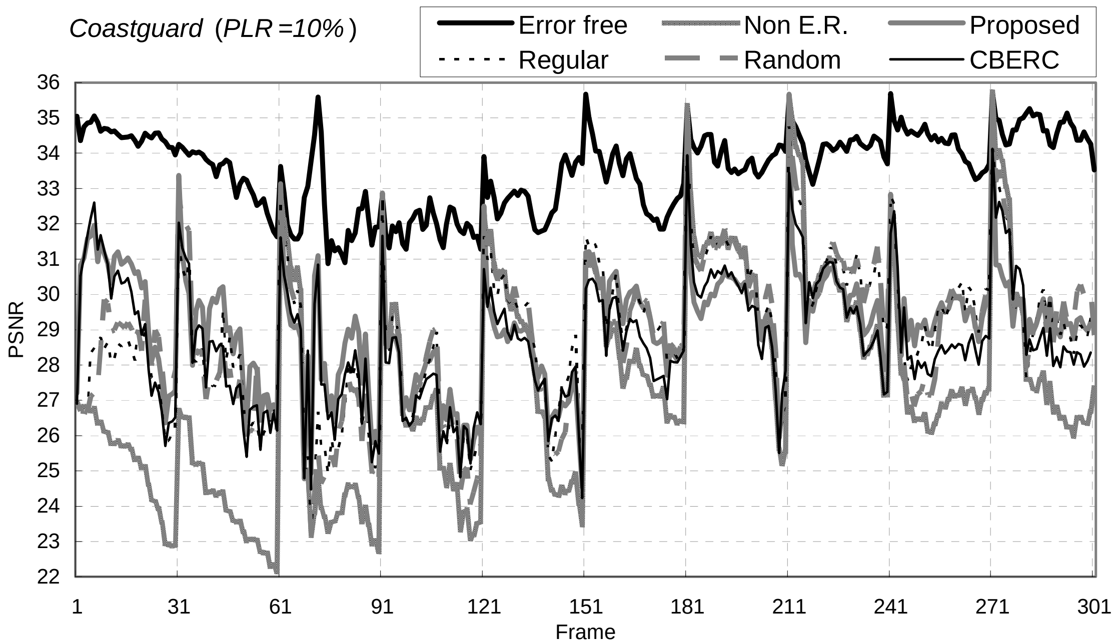


(a) PLR = 10%

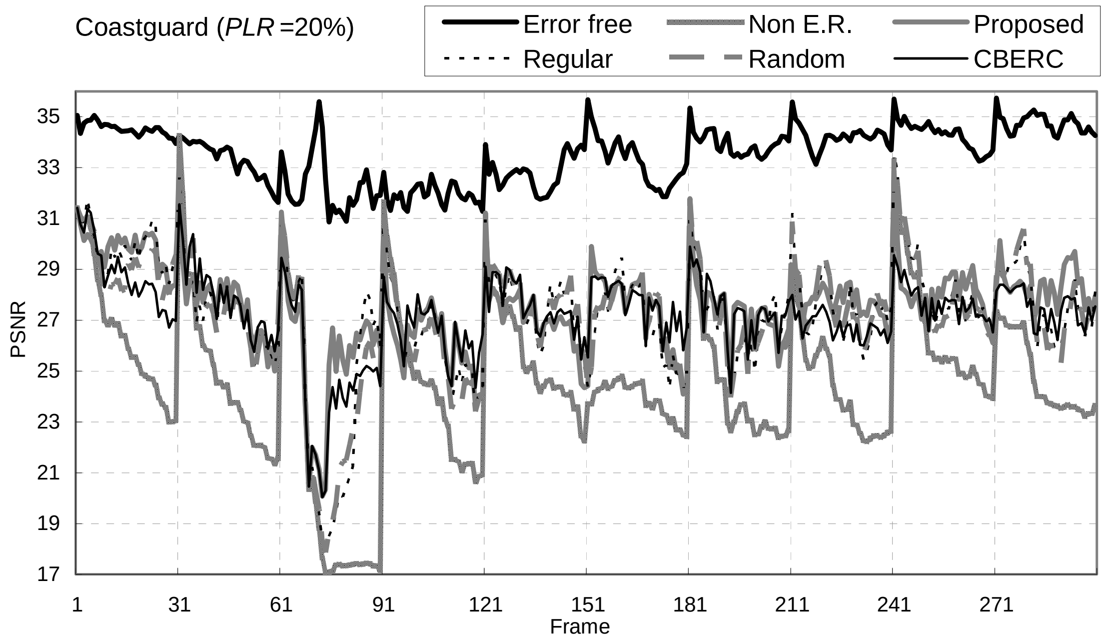


(b) *PLR* =20%

Fig. 8. Frame-by-frame PNSR performance comparison using four intra-refresh methods under four different channel conditions (Foreman).

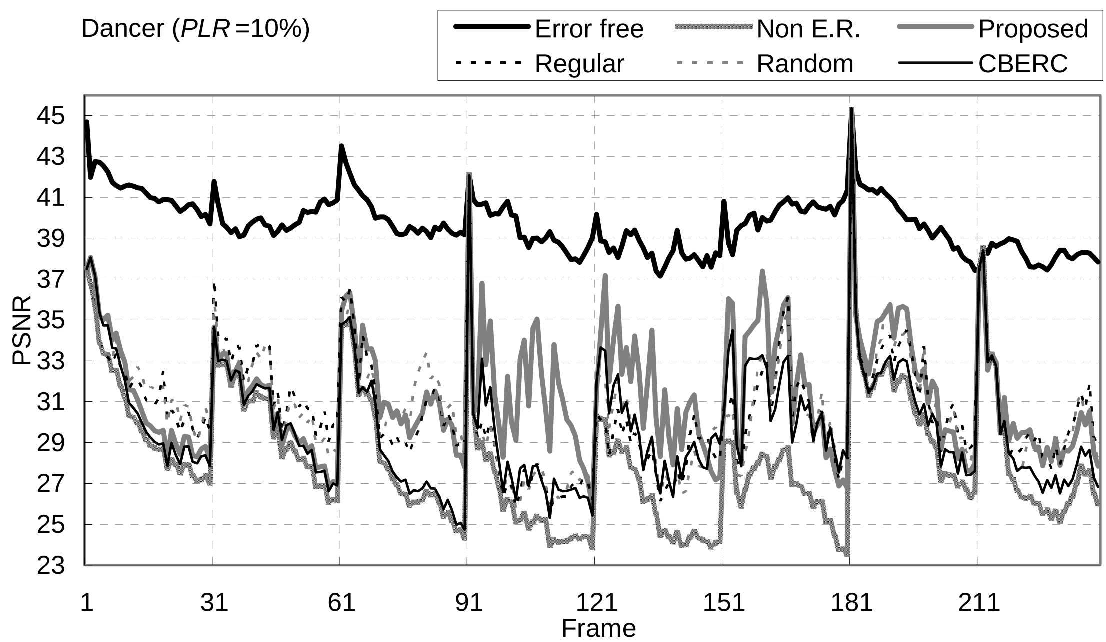


(a) *PLR* = 10%

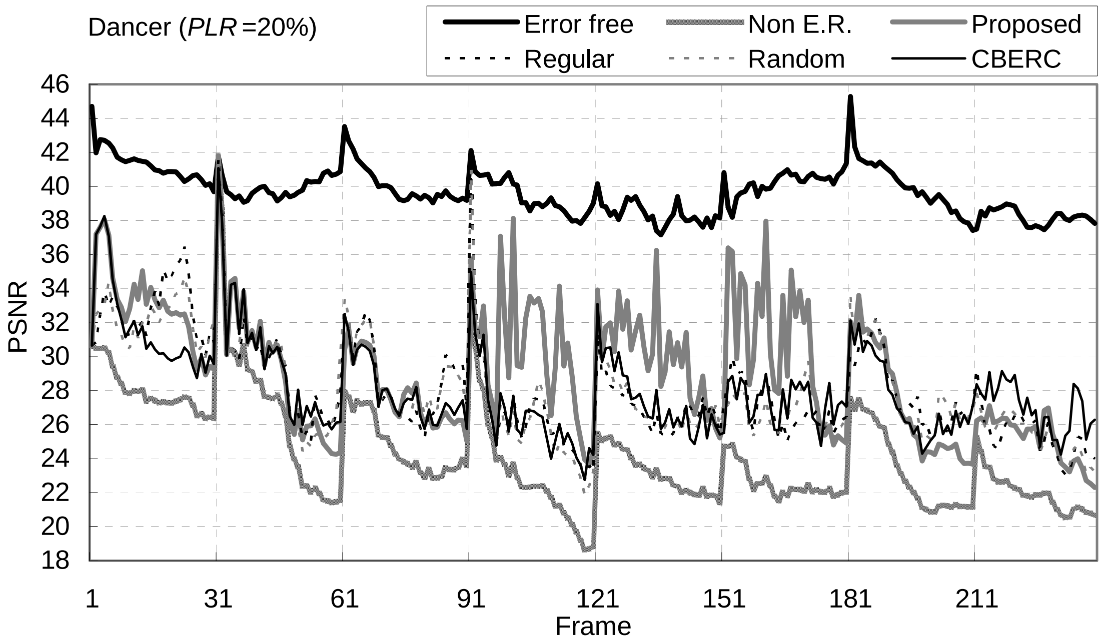


(b) *PLR* =20%

Fig. 9. Frame-by-frame PNSR performance comparison using four intra-refresh methods under two different channel conditions (Coastguard).

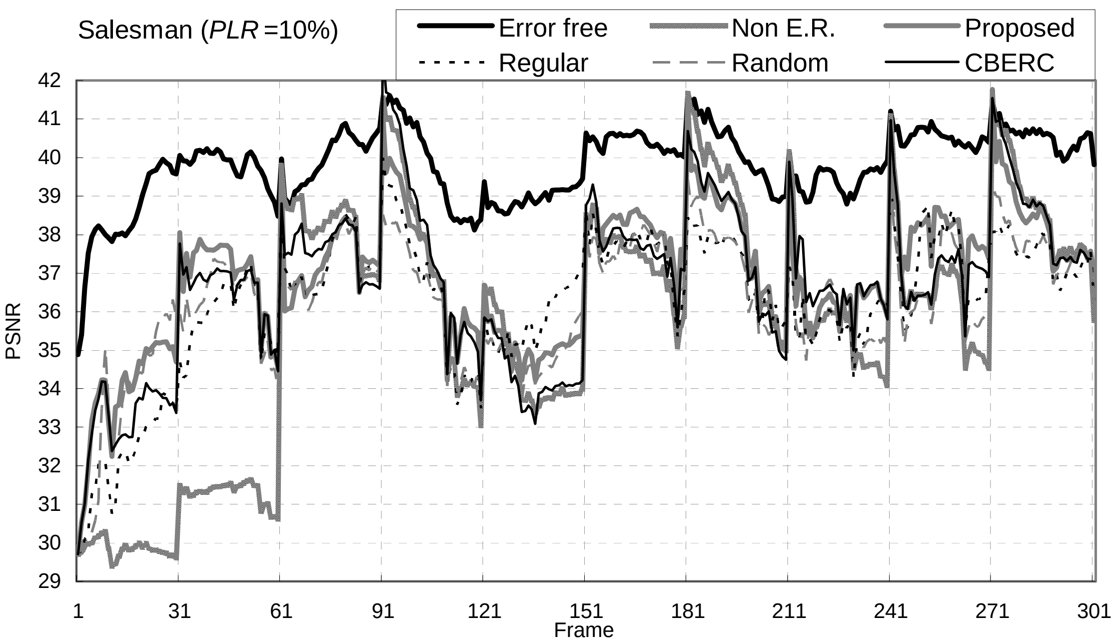


(a) *PLR* = 10%

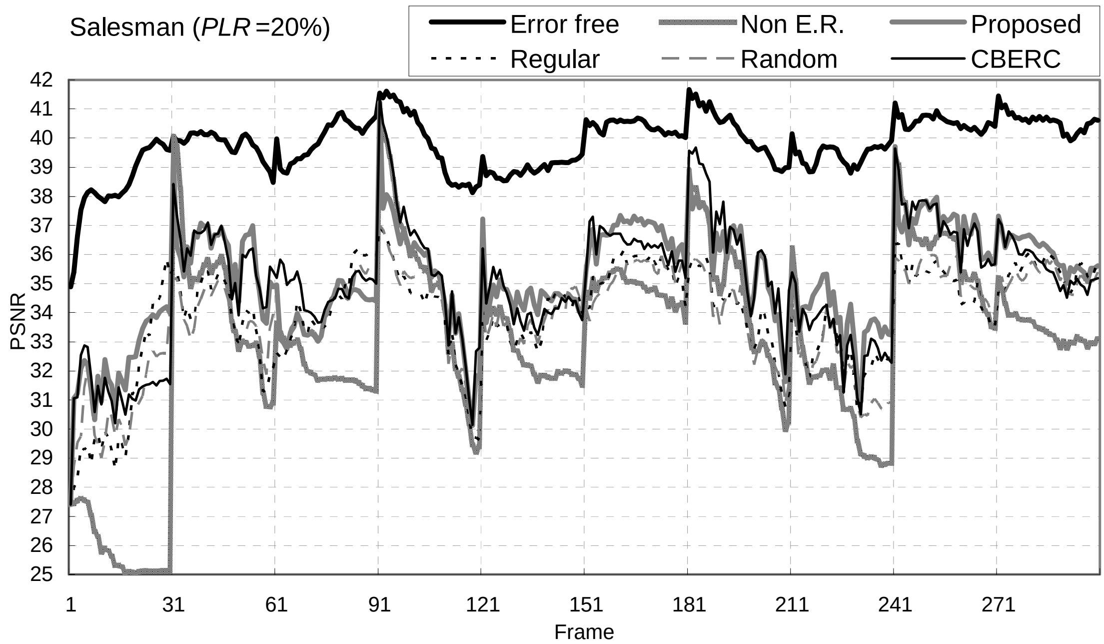


(b) *PLR* = 20%

Fig. 10. Frame-by-frame PNSR performance comparison using four intra-refresh methods under two different channel conditions (Dancer).

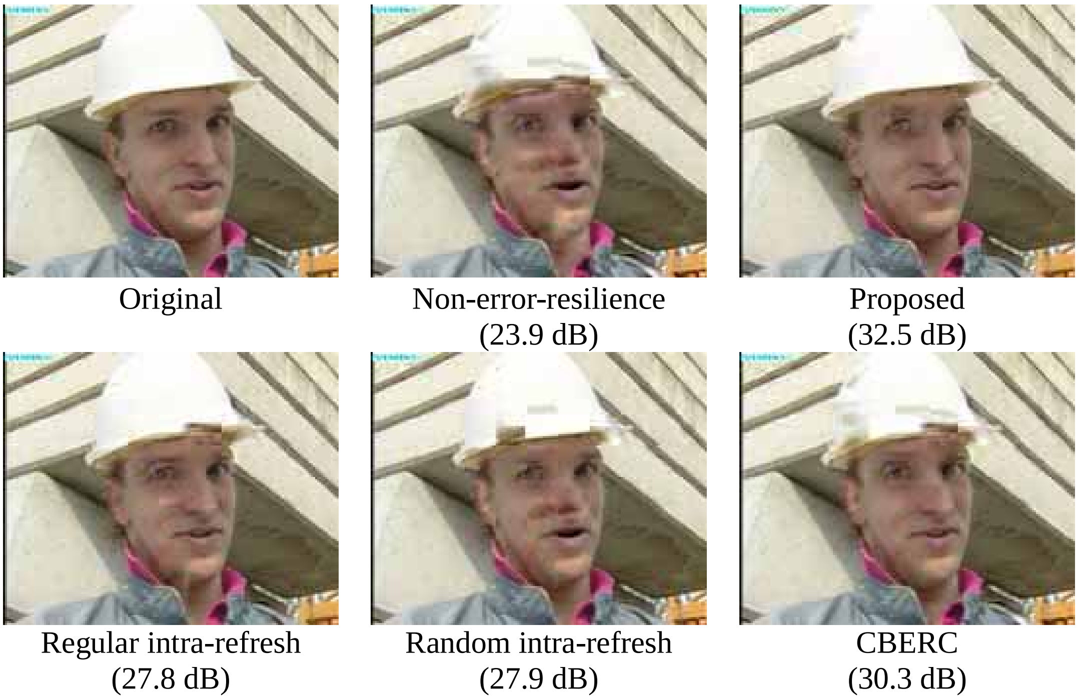


(a) *PLR* = 10%

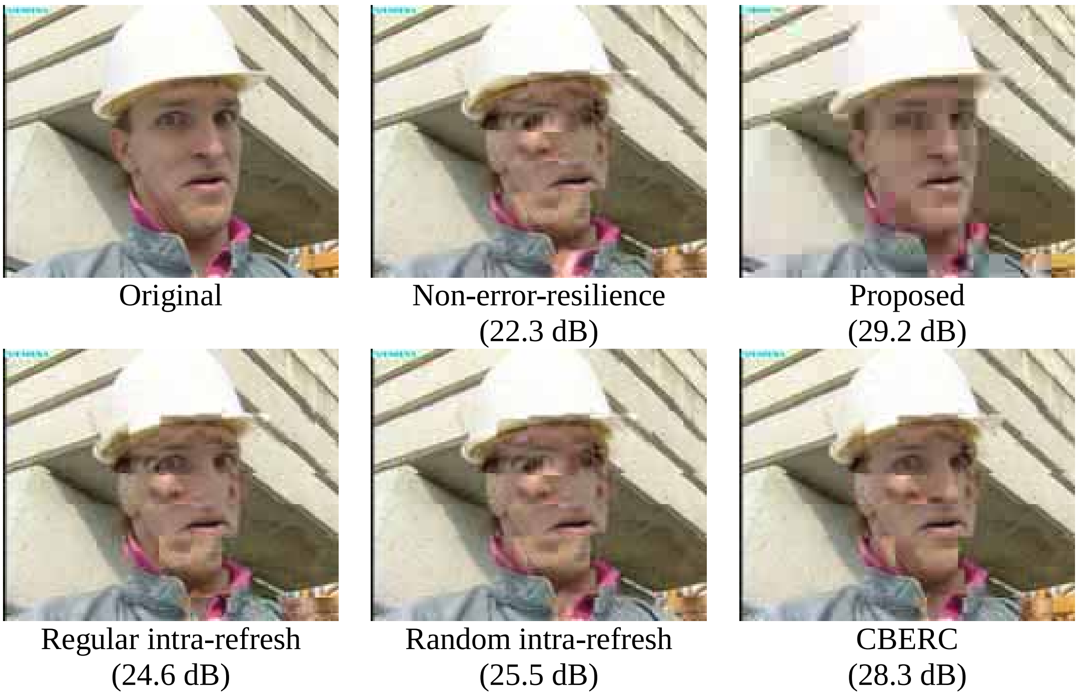


(b) *PLR* = 20%

Fig. 11. Frame-by-frame PNSR performance comparison using four intra-refresh methods under two different channel conditions (Salesman).



(a) Foreman, frame 108 with packet loss rate 10%



(b) Foreman, frame 98 with packet loss rate 20%

Fig. 12. Video snapshots for subjective quality comparison between five schemes with *PLR* = 10% and 20% (Foreman).


Original


Non-error-resilience (24.4 dB)


Proposed (30.2 dB)


Regular intra-refresh (29.5 dB)


Random intra-refresh (28.6 dB)


CBERC (28.5 dB)

(a) Coastguard, frame 44 with packet loss rate 10%


Original


Non-error-resilience (25.4 dB)


Proposed (27.6 dB)


Regular intra-refresh (26.5 dB)

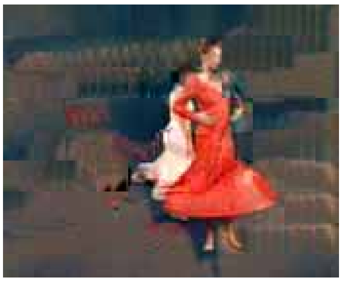

Random intra-refresh (26.6 dB)


CBERC (26.8 dB)

(b) Coastguard, frame 215 with packet loss rate 20%

Fig. 13. Video snapshots for subjective quality comparison between five schemes with *PLR* = 10% and 20% (Coastguard).

Original

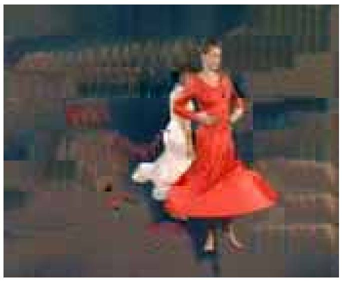


Non-error-resilience
(25.3 dB)

Proposed
(35.0 dB)

Regular intra-refresh
(27.9 dB)

Random intra-refresh
(27.6 dB)



CBERC
(28.8 dB)

(a) Dancer, frame 107 with packet loss rate 10%

Original

Without error-resilience
(21.2 dB)

Proposed
(34.1 dB)

Regular intra-refresh
(25.9 dB)

Random intra-refresh
(24.8 dB)

CBERC
(26.0 dB)

(b) Dancer, frame 112 with packet loss rate 20%

Fig. 14. Video snapshots for subjective quality comparison between five schemes with *PLR* = 10% and 20% (Dancer).

Original

Without error-resilience (31.3 dB)

Proposed (37.8 dB)

Regular intra-refresh (35.7 dB)

Random intra-refresh (36.1 dB)

CBERC (37.0 dB)

(a) Salesman, frame 37 with packet loss rate 10%

Original

Non-error-resilience (32.8 dB)

Proposed (36.9 dB)

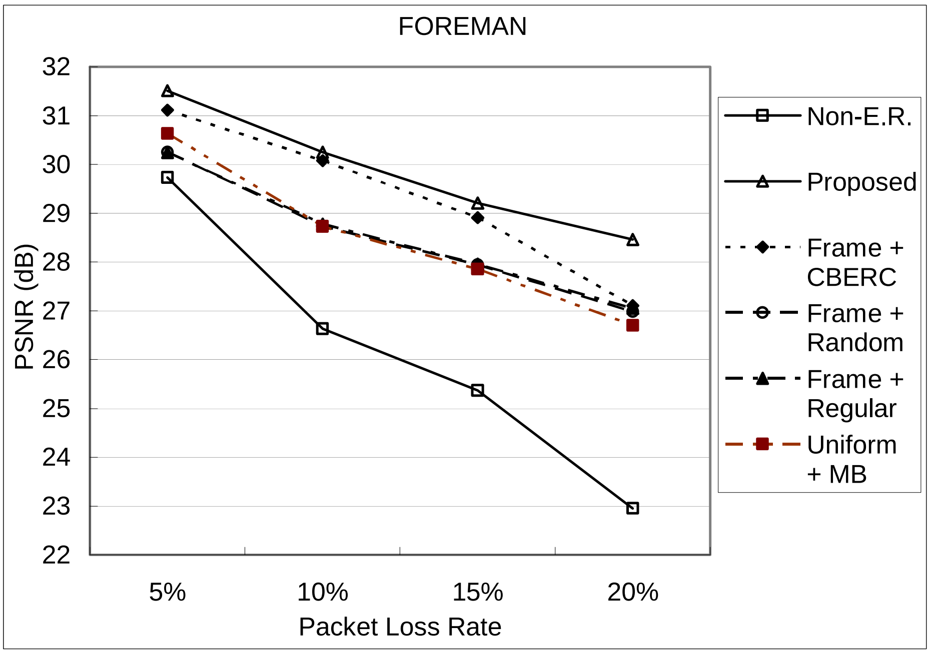
Regular intra-refresh (34.1 dB)

Random intra-refresh (33.6 dB)

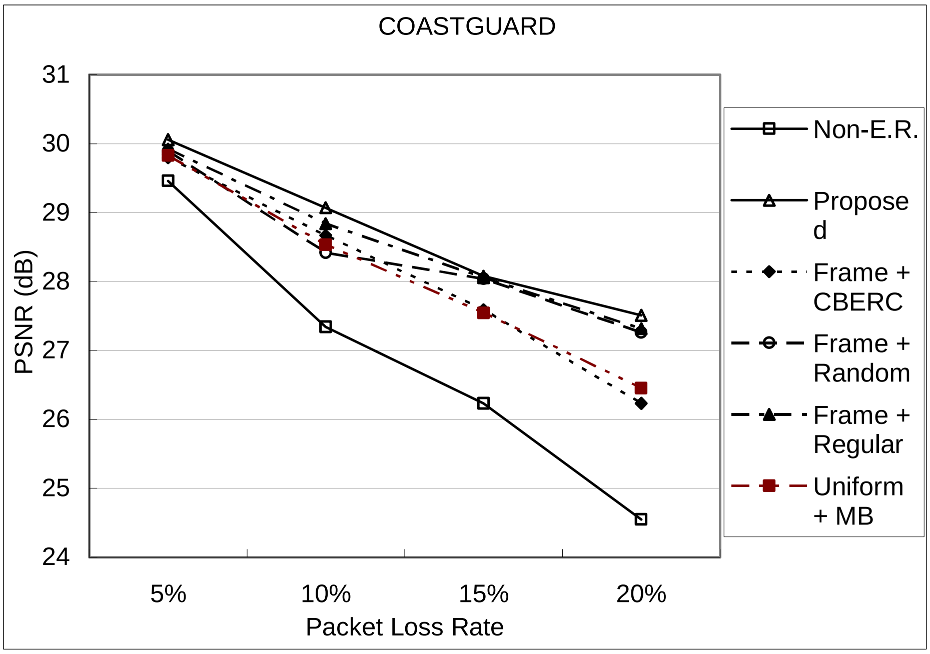
CBERC (36.2 dB)

(b) frame 54 with packet loss rate 20%

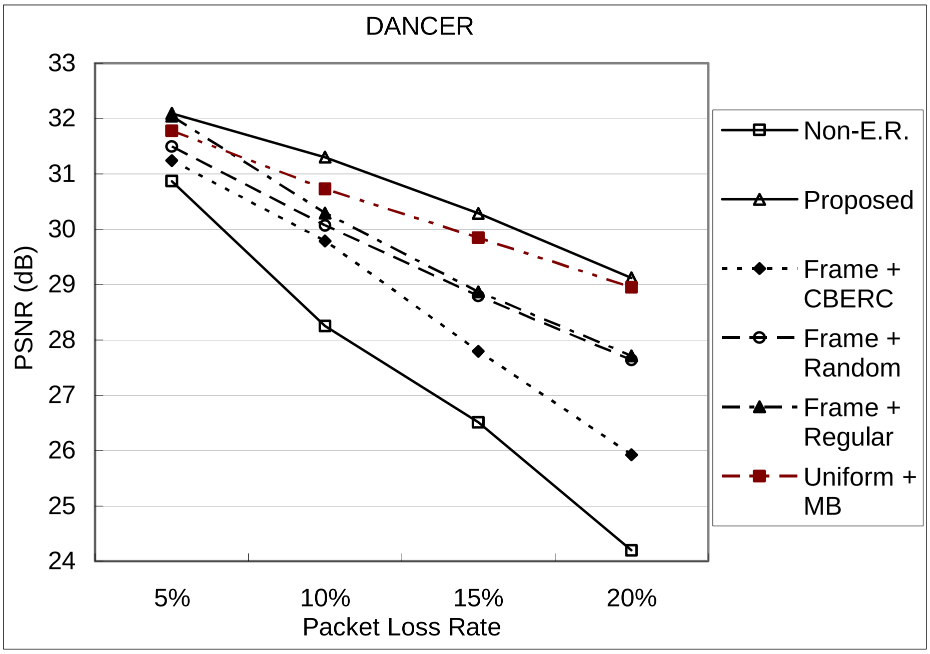
Fig. 15. Video snapshots for subjective quality comparison between five schemes with *PLR* = 10% and 20% (Salesman).



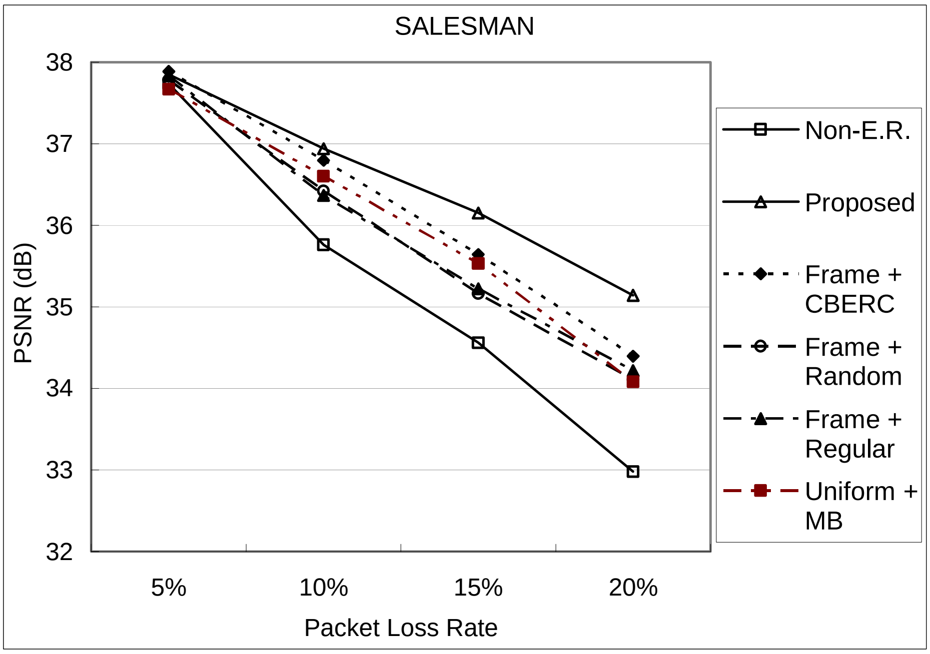
(a) Foreman



(b) Coastguard



(c) Dancer



(d) Salesman

Fig. 16. Average PNSR performance comparison using various combinations of frame-level and MB-level intra-refresh distribution methods under four different channel conditions.