Error robust H.263 video coding system

Tien-Hsu Lee and Pao-Chi Chang*
Department of Electrical Engineering, National Central University

ABSTRACT

This paper presents an error resilient H.263 video compression system over noisy channels. We develop a video segment regulation algorithm at the decoder to efficiently identify and correct erroneous start codes and block addresses. In addition, a parity-embedded error detection technique is also implemented to enhance the error detection capability of the decoder at the macroblock-layer. After performing above two approaches, the decoder can report the accurate addresses of detected corrupt blocks back to the encoder via a feedback channel. With these negative acknowledgments, the precise error tracking algorithm is developed at the encoder to precisely calculate and trace the propagated errors for INTRA refreshing the contaminated blocks. Simulation results show that the proposed system yields significant video quality improvements over the motion compensated concealment by PSNR gains of 4 to 6 dB at bit rate around 32 kbps in error-prone DECT environments. In particular, this system complies with the H.263 standard and has the advantages of low memory requirement and computation complexity that are suitable for practical real-time implementation.

Keywords: H.263 video coding, noisy channel, resynchronization, error detection, error tracking

1. INTRODUCTION

With the growing popularity of computers and networks, the multimedia communication services are getting increasingly important. In particular, the demand for digital video applications such as the video conferencing and video telephony is increasing rapidly [1]. However, it is nearly impossible to transmit the huge video data without compression for the real-time communications over bandwidth-limited networks [2]. Consequently, the video compression techniques become increasingly essential and gain much more attention in recently.

Much effort has been made by ITU-T (International Telecommunication Unit) to standardize the H.263 system for the low bitrate video communication [3]. Not only in the public switched telephone network (PSTN), but also in the wireless environment H.263 is being considered for the transmission of low bit rate video since the mobile communication becomes a more important part of daily life [4]. However, wireless channels are subject to the multi-path fading, which results in bursty errors. Such errors may cause the error propagation effects and severely degrade the perceived video quality. This is because H.263 uses the motion-compensated predictive coding, differential coding of motion vectors, and variable length entropy coding, which are extremely vulnerable to transmission errors.

Several techniques, such as the unequal error protection [5], ARQ (Automatic Retransmission reQuest) [6], error concealment [7-9], and forced INTRA updating [10], have been proposed to limit the effects of error propagation. The unequal error protection is not compatible to the syntax of the coding standard while the ARQ is usually not suitable for the real-time video communication. The performance of the error concealment generally relies on not only the validity of neighboring blocks, but also the accuracy of the error detection. The forced INTRA updating can effectively stop the error propagation but with the penalty of overshoot in the bit rate.

The enhanced versions of H.263 such as H.263+/++ and H.26L are now investigated [11]. It is especially worth to note that an error tracking strategy is described in Appendix I of H.263+ [12] to combat the serious spatial-temporal error propagation effects if the erroneous macroblocks (MBs) are reported via a feedback channel to the encoder. Rapid and efficient error recovery can be achieved by INTRA refreshing those estimated corrupt MBs. As a result, for eliminating the error propagation effects, the error concealment needs accurate corrupted locations (i.e., great error detection

* pcchang@ee.ncu.edu.tw; phone 886 3 422-7151 ext. 4466; fax 886 3 425-5830; http://vaplab.ee.ncu.edu.tw;
Department of Electrical Engineering, No. 300, Jung-Da Rd., Jung-Li City, Tao-Yuan, Taiwan 320, R.O.C.

capability) and the INTRA refreshing method needs accurate error tracking capability. Moreover, the start codes in the H.263 bit stream syntax, which inherently provide the resynchronization functionality for error handling, may also cause significant error damage if they are incorrectly decoded. Error protection or error correction strategies should be developed to improve the robustness of such important information.

In this paper, our main objective is to develop a robust H.263 video coding system that could be implemented in an error-prone environment, such as the wireless network. The proposed system consists of three essential and novel techniques, including the video segment regulation (VSR) [13], the parity embedded error detection (PEED) [14], and the precise error tracking (PET) [15]. The VSR and PEED perform at the decoder and the PET does at the encoder. The VSR corrects erroneous resynchronization codes and GOB/MB addresses. The PEED, as well as the VSR, provides the accurate negative acknowledgements (detected error locations) to the PET via a feedback channel. Finally, the core PET stops the spatial-temporal error propagation by INTRA coding the estimated corrupt MBs.

This paper is organized as follows. In the next section, we introduce the architecture of the proposed error resilient H.263 coding system. Section 3 describes the three novel techniques, VSR, PEED, and PET, applied to this system in detail. Subsequently, the simulation results of our system are demonstrated in section 4. Finally, section 5 concludes our work.

2. SYSTEM ARCHITECTURE

The decoder and encoder schemes we propose are depicted in Fig. 1. Robust error detection and error concealment must rely on correct identification of each GOB (or slice) location and new picture start position. For this reason, we perform the proposed video segment regulation technique, i.e., the VSR, at the decoder to identify and correct erroneous start codes and block addresses in headers. A video segment can be a GOB in H.263 or a slice in H.263+. The decoder preprocesses the received compressed bit stream, which is temporarily stored in the receiver buffer, to search for the start codes and their corresponding encoded video segments. Every start code number is examined by a checking procedure to see if it is a correct one. Then, the regulation procedure is performed on all erroneous ones by minimizing the Hamming distance distortion measure. Therefore, errors occurring in start code numbers can be mostly corrected and consequently all GOB (or slice) locations are uniquely identified.

At the decoder, we utilize the PEED technique to find transmission errors occurred in video data segments. At the encoder, the requisite parity-check codes for all MBs in the previous frame are embedded into the motion vectors and the quantized residual DCT coefficients of the current encoding frame bitstream. Accordingly, the decoder can effectively manipulate these embedded bits to detect the actual error occurring MB-locations for enhancing the error detection capability. Finally, these negative acknowledgements (NACKs) should be sent back to the encoder for performing the error tracking or to the decoder for performing the error concealment.

With the analysis of temporal dependencies of MBs in successive frames, the feedback NACKs are utilized by the encoder for reconstructing the error propagation effects at the encoding end and selecting severely affected regions to be INTRA refreshed. Our proposed precise error tracking strategy, i.e., the PET, uses the pre-stored motion vectors (MVs) and traces the motion dependency for each current encoding pixel backward to the previous unsuccessfully decoded frame informed by the NACK. The encoder thus can exactly evaluate how severely every MB in the current encoding frame is affected by those impaired areas. Finally, all or parts of the contaminated MBs can be selected to refresh by the INTRA-mode coding. It is especially worth to note that this algorithm is able to track the actual error propagation and completely terminate it. The earlier the feedback-transmission (i.e., the NACK) arrives, the sooner the error-propagation stops.

To sum up, the proposed system integrates these three novel techniques and provides a complete solution to the H.263 standard-compatible video coding used in the error-prone environment. It can effectively enhance the error detection capability and stop the error propagation effect. In practice, the overall scheme is especially suitable for the real-time implementation due to its low memory requirement and computation complexity.

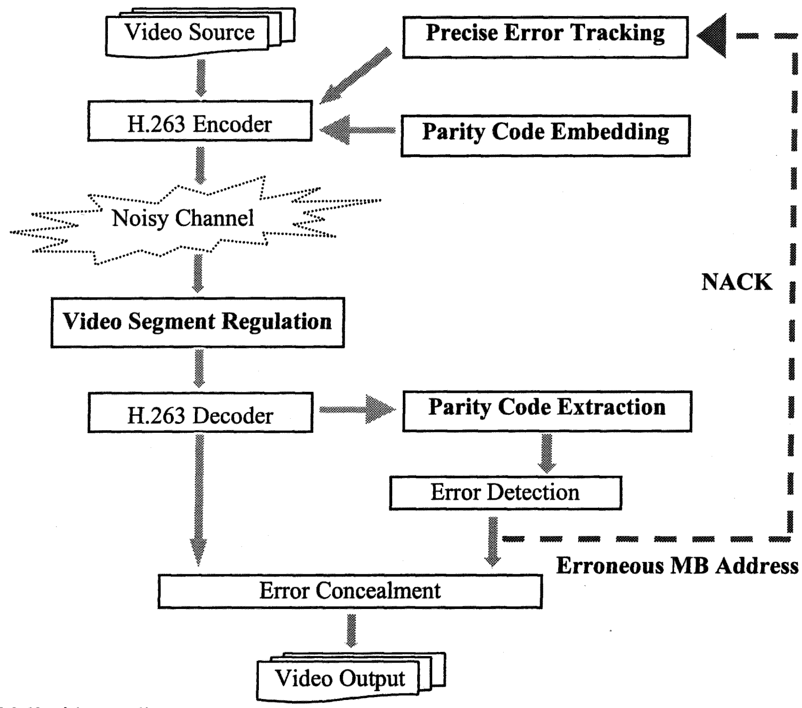

Figure 1: Error robust H.263 video coding system.

3. KEY TECHNIQUES

The proposed system consists of three essential and novel techniques, including the video segment regulation, parity embedded error detection, and precise error tracking algorithms. The details of each approach are described as follows.

3.1 Video segment regulation

Due to the use of variable length coding (VLC), the erroneous compressed bit stream usually can not be decoded correctly until the next resynchronization point, i.e., the next start code position. Consequently, the start code provided by the H.263 standard plays an important role for the error detection and error recovery in error-prone environments. The start codes only exist in the picture and GOB (or slice) layers. In other words, the minimum resynchronization unit is one GOB (or slice). A single bit error may corrupt an MB and the subsequent MBs in the same GOB (or slice).

Based on the H.263 standard, there exists a tradeoff in the choice of the occurrence frequency of start codes. The less the start codes are used (or the longer the slice length is used), the longer the errors are propagated. However, frequent start codes result in the substantial overhead in the bit rate and the increased occurrence probability of erroneous start codes. A missing or misinterpreted start code generally results in much more serious image degradation than the errors in image data. For example, one bit error may result in the entire GOB 5 of a frame falsely decoded into GOB 1 and thus both GOBs are corrupted. Particularly, a missing or fake picture start code (PSC) may come out with the incorrect number of decoded frames and the inaccurate playback time. Furthermore, the accuracy of the error tracking also relies on the correct feedback addresses of corrupted MBs.

To solve above problems, we develop the video segment regulation algorithm to correct erroneous start codes and block addresses. The flow chart is depicted in Fig. 2. The proposed algorithm mainly comprises two parts, the erroneous video segment allocation and rearrangement. In the procedure of erroneous video segment allocation, the decoder first preprocesses the received bit stream, which is temporarily stored in the receiver buffer, to search for the start codes and their corresponding encoded video segments. According to the GOB structure syntax, a start code number is defined as the GOB start code (GBSC) concatenated with the following group number (GN). If the slice structured mode is in use, the start code number, which contains the slice start code (SSC) and the MB address (MBA), is given in an increasing order. A start code is then marked as in a correct video segment if its number is in the consecutive order with the preceding one and the succeeding one. In this case, the decoder continues decoding this video segment as normal. Otherwise, the erroneous video segments are kept in the receiver buffer until the next correct start code number is identified.

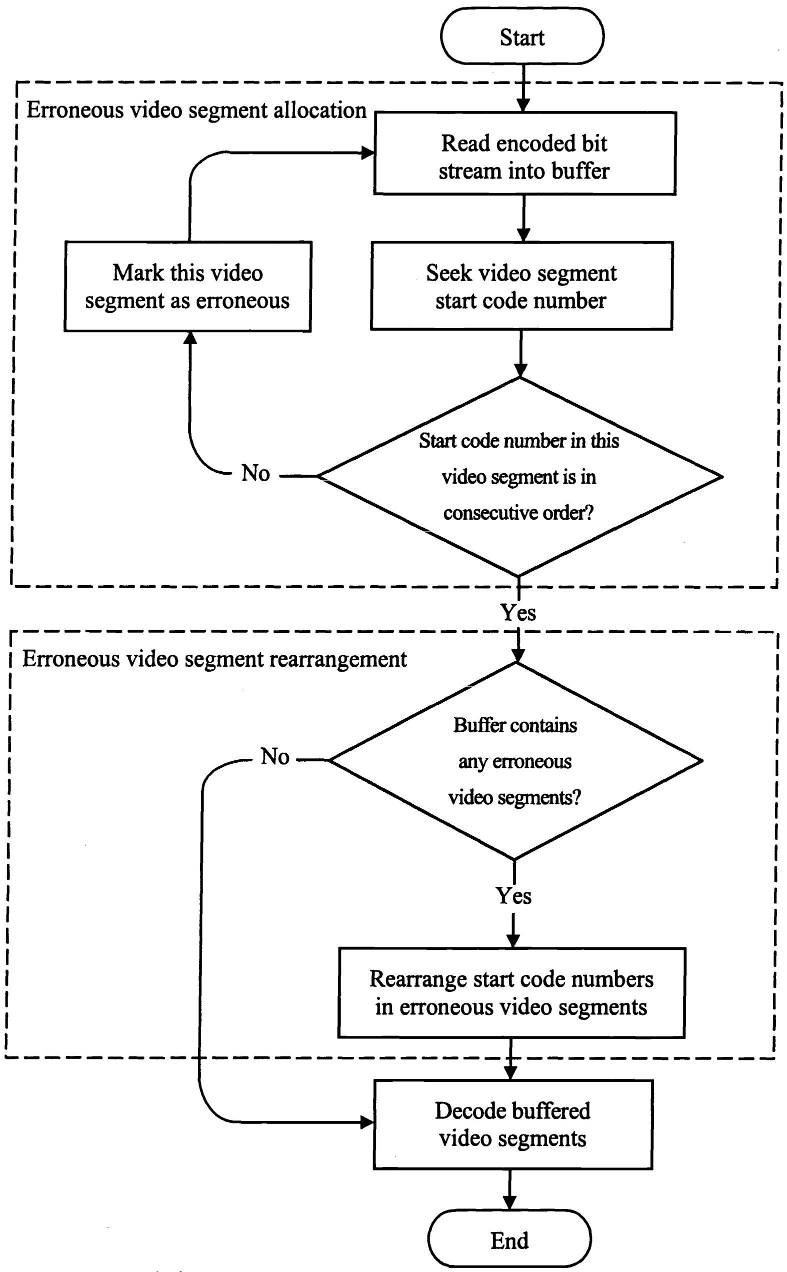

Figure 2: Flow chart of video segment regulation.

After the erroneous video segment allocation, the rearrangement procedure is performed on these erroneous video segments. Here, we denote the number of detected erroneous video segments between two correct ones as N_{found} and the number of desired consecutive video segments between two correct ones as N_{need}. Every erroneous segment is classified as one of the following three cases and the corresponding rearrangement procedure is fulfilled.

(a) Forced start code assignment ($N_{found} = N_{need}$): All erroneous start codes are forcedly changed to the corresponding desired start code numbers that are in the consecutive order.
(b) Lost start code reconstruction ($N_{found} < N_{need}$): The lost start code is reconstructed by searching the erroneous video segments for the bit pattern that has the minimum Hamming distance to the desired one.
(c) Extra start code erasure ($N_{found} > N_{need}$): This case rarely happens because the H.263 start code length is 17 bits long (0000 0000 0000 0000 1, without GN or MBA). If it happens on occasion, the start codes with the minimum Hamming distance to the desired start code numbers are changed forcedly. Then the rest start codes with their video segments are deleted from the bit stream.

After performing the video segment regulation, all GOB (or slice) locations are uniquely recognized and the video sequence can be decoded accordingly. Note that this regulation algorithm cannot be applied to the arbitrary slice ordering submode in H.263+ if the slice structured mode is in use. Also note that the picture start code, whose MBA is '0', can also be included in the regulation process. Therefore, this process regulates not only the slice order, but also the picture synchronization.

3.2 Parity embedded error detection

Due to the use of VLC, a single error in the H.263 video bit stream may cause serious error propagation and quality degradation. Many error recovery techniques have been applied to the H.263 video coding. Most importantly, the efficiency of the error recovery relies on the capability of the error detection. Consequently, we implement an effective error detection technique, called the parity-embedded error detection, for the H.263 compatible video coding system. The requisite parity-check information is embedded in the encoded bit stream, which could be easily extracted by the decoder for detecting the actual error occurrence locations. At the encoder, the parity-check codes for all macroblocks in the previous frame are embedded into the motion vectors and the quantized residual DCT coefficients of the current encoding frame. Then the decoder can effectively manipulate these embedded bits to enhance the error detection capability at the MB-layer. The overall computation complexity is considerably low and the additional cost is only one-frame coding delay and little bitrate increase.

3.2.1 Parity code embedding and extraction

What we want to embed is the parity-check code for each MB. It is defined as the modulo-2 summation of the encoded bit stream that belongs to the corresponding MB. Thus, we need to embed 99 parity-check codes (bits) for a QCIF-format (176 x 144) video. The generated parity-check codes of the current frame will be embedded into the next frame. Two strategies can be adopted to embed parity-check codes.

(a) Data embedding in motion-compensated residual DCT coefficients (DERC): As for an embedded bit '0' ('1'), if the summation of all non-zero residual DCT coefficients of an MB is odd (even), one quantized value of such coefficients belonging to this MB should be added or subtracted by one. The coefficient should be selected to the one that has the minimum new quantization error. Therefore, such a change will not degrade the visual quality and increase the bit rate too much.
(b) Data embedding in motion vectors (DEMV): This strategy is originally proposed by Song and Liu in [16]. They embed data by changing the half-pixel motion estimation. We have modified the DEMV method by adding a constraint. When an MB is not coded, i.e., COD = 1, the parity bits will not be embedded into this MB. This constraint will avoid the DEMV from significantly increasing the bit rate.

DERC and DEMV may be used together or separately. At the decoder, the parity-check code for each MB in the current frame can be easily extracted from decoding the bit stream of the next frame according to the way they had been embedded at the encoder. The procedure of parity extraction is described as follows.

Step 1: If the received INTER MB is not coded (COD = 1) or the received MB is INTRA-coded, no information bit is hidden in this MB.
Step 2: If the number of nonzero coefficients in this MB is greater than a pre-determined threshold, then we check the sum of all quantized values. If the sum is odd, the information bit is 1; otherwise, the information bit is 0.
Step 3: For an INTER MB (COD = 0), a motion vector exists. Check the x and y components of this motion vector separately and obtain two information bits accordingly. If the decoded motion vector component is an integer,

the hidden information bit is 0; otherwise, if the motion vector is in half-pixel positions, the information bit is 1.

Step 4: Arrange the extracted information bits in order until all the information bits have been extracted. For example, there are 99 MBs in a QCIF format picture. The total parity check codes thus have 99 bits.

3.2.2 Error detection

After the embedded bit extraction, the correctness of each MB is checked. Ideally, the error detection is accomplished by comparing the extracted information bits from the current frame with the parity check code of each MB in the previous frame. If they are different, errors exist. However, there is an exception. If an error occurs in frame *N*, therefore the extracted information bits (in frame *N*) are disturbed and incorrect. We then may mistakenly think that frame *N-1* is erroneous because its MB parity bits do not match the extracted bits. Fortunately, the H.263 decoder itself exists several inherent error detection capabilities that can help the operation of PEED mutually. They, defined as the default error detection (DED), include:

- An invalid VLC codeword is found.
- The total number of decoded MBs (blocks) within a GOB (MB) is illegal.
- The number of decoded DCT coefficients within a block is larger than 64.
- The length of a GOB (the minimum resynchronization unit) is illegal. For example, the extra codeword or premature end is found before the start code.
- Impossible decoded values are found. For example, the QUANT code or MV value is out of range.

3.3 Precise error tracking

NACKs with the information on the unsuccessfully decoded image blocks are sent back to the encoder via a feedback channel. Once a NACK is received, the encoder performs the error tracking to determine whether the current encoding MB is contaminated by the erroneous MBs in the past frames. If this happens, this MB becomes a candidate for INTRA-refresh coding. The dependencies of MBs in successive frames are essential to the error tracking. The MV produced in the motion estimation indeed provides adequate information for accurately tracing the error propagation in the encoder.

The example in Fig. 3 illustrates how to execute the precise error tracking for the three pixels *Q*, *R*, and *S* by employing the *pixel-based backward motion dependencies* in the QCIF format video. Any pixel's motion dependency can be found by tracing back the MV of the MB it belongs to. We suppose that the prediction of MB 25 in frame N is obtained from MBs 25 and 26 in frame N-1. MB 26 of frame N-1 refers to MBs 15, 16, 26, and 27 in frame N-2. MB 25 of frame N-1 is coded in INTRA mode. If the encoder receives a NACK, which indicates an error occurred at MB 15 of frame N-2, while encoding frame N. At that moment, we can backward trace each pixel in MB 25 of frame N along the corresponding paths, i.e., the corresponding MVs, to see if it refers to the erroneous area, i.e., MB 15 of frame N-2. As shown in Fig. 3, pixel *Q* is then determined to be a contaminated pixel while pixel *R* is not. Likewise, the backward motion dependency structure for each MB can be built as in Fig. 3 and the error tracking procedure is performed for all pixels in frame N. It is especially important to note that there exists no MV for an INTRA MB, such as MB 25 of frame N-1. In such a case, pixel *S* in Fig. 3 is certainly claimed to be clean.

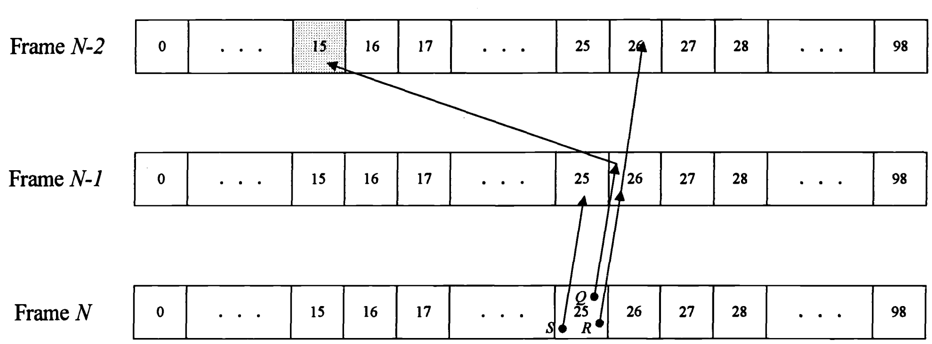

Figure 3: Illustration of precise error tracking.

Although the INTRA-mode coding terminates the temporal error propagation, it usually generates a higher bit rate, too. If the rate increased by the INTRA-mode coding is higher than what we can afford, we need to limit the number of INTRA MBs in a frame. For instance, we can select only a fixed number of MBs to perform INTRA refreshing. Of course, in such cases the error propagation may not be stopped completely. On the other hand, in spite of the fact that this algorithm requires tracing the motion dependency for each pixel, it actually exhibits very low complexity because only simple additions are needed. By setting a maximum round-trip delay, the memory requirement is also low since all we need to store is the required MV information, which has been generated in the motion estimation procedure already.

In summary, the PET utilizes the feedback NACKs to reconstruct the error propagation at the encoder. By utilizing the MVs and by tracing the backward motion dependency for each pixel, contaminated MBs can be accurately tracked and the error propagation is terminated through INTRA coding the affected MBs. It is worth to note that in general the data rate of NACKs for sending back the range of erroneous MBs is extremely low. If the NACKs are unfortunately corrupted by errors, we can assume that the retransmission or FEC protection is applied by external means. Thus, correct NACKs are assumed in the following simulations of this paper (i.e., the feedback channel is noiseless).

4. SIMULATION RESULTS

Two video sequences, *Foreman* and *Salesman*, are tested in the simulations. They are QCIF (176 x 144) format, reference frame rate at 30 Hz, and 400 frames in length. These sequences are compressed by H.263 without all negotiable coding optional modes and the rate control. The encoded frame rate is kept to be 10 Hz. By default, the motion compensated concealment (MC) is used for the error recovery of detected erroneous MBs at the decoder. Here, the MC makes use of the average MV of the upper and left neighboring MBs as the erroneous MB's. All numerical results are obtained from the average over 40 simulation iterations for different offsets in the bit error patterns.

These two video sequences are tested in nine simulated DECT channels, which are generated by a correlated Rayleigh fading model for the DECT system at 14-km/h speed [17], [18]. The characteristics of nine channels are listed in Table 1. The round-trip delay for receiving a NACK is assumed 300 ms.

Table 1: Characteristics of DECT channels.

Es/No (dB)	Average Bit Error Rate (%)
14	9.786E-01
16	6.328E-01
18	4.095E-01
20	2.644E-01
22	1.703E-01
24	1.091E-01
26	6.995E-02
28	4.422E-02
30	2.777E-02

The PSNR simulation results for sequence *Foreman* and *Salesman* are compared in Fig. 4 and Fig. 5, respectively. Here, the DERC and DEMV are utilized simultaneously. We can find that the PEED indeed detects more accurate error locations than the DED does. Therefore, the performances of the PET and MC are also improved and the video quality is further enhanced. In fact, from the work presented in [14], the PEED strategy is able to detect errors with extra 10 % ~ 30 % accuracy rather than the DED provides at the bit error rate $10^{-2} \sim 10^{-4}$.

Finally, we demonstrate and compare the video quality by employing different strategies for sequence *Foreman* in the DECT channel at Es/No = 20 dB. All parameters and notations are the same as the mentioned above. The resultant encoded bitrates are all constrained to about 75 kbps. The simulation results are shown in Fig. 6. It is obvious to find that the proposed system yields the best subjective visual quality. More specifically, the average PSNR for only DED is 20.6 dB. As for VSR with PEED, the average PSNR is 21.3 dB. Furthermore, by executing PET, the average PSNR will be significantly improved to 25.3 dB.

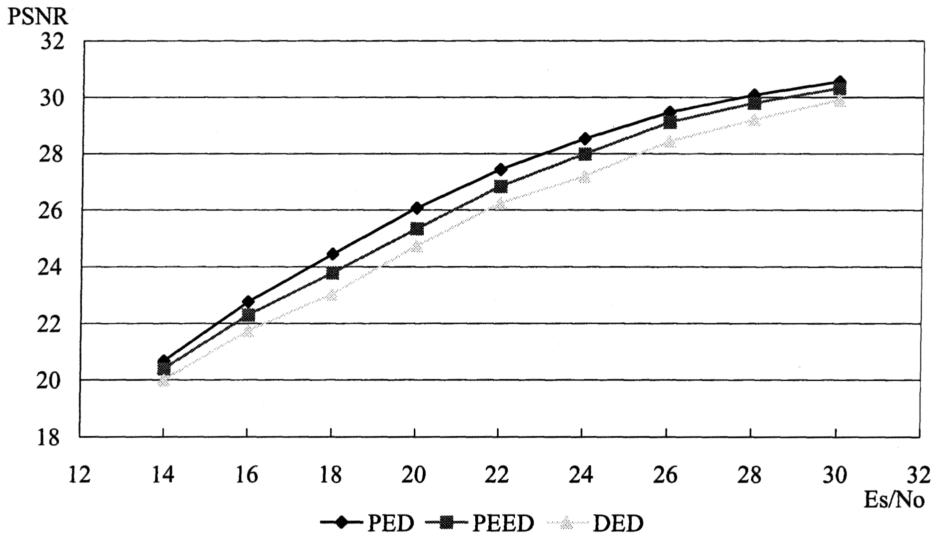

Figure 4: PSNR simulation results for sequence *Foreman* in DECT channels.

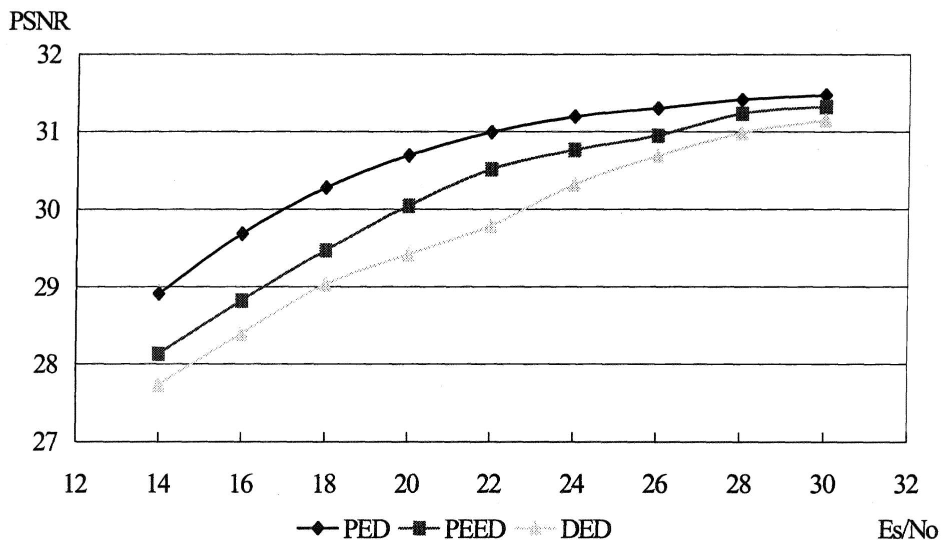

Figure 5: PSNR simulation results for sequence *Salesman* in DECT channels.

(a) Frame 150 (b) Frame 300

(c) Frame 150 (d) Frame 300

(e) Frame 150 (f) Frame 300

Figure 6: Video quality comparisons for sequence *Foreman* in the DECT channel at Es/No = 20 dB: (a)(b) DED, (c)(d) VSR + PEED, and (e)(f) VSR + PEED + PET (i.e., the proposed system).

5. CONCLUSION

This paper presents a robust H.263 video-coding scheme that could be implemented in an error-prone environment, such as the wireless network. The proposed scheme comprises three essential and novel techniques, including the video segment regulation, parity embedded error detection, and precise error tracking. The VSR and PEED perform at the decoder and the PET does at the encoder. The VSR corrects most erroneous resynchronization codes and GOB/MB addresses. The PEED, as well as the VSR, provides accurate negative acknowledgements (detected error locations) to the PET via a feedback channel. Eventually, the PET stops the spatial-temporal error propagation by INTRA refreshing the estimated corrupt MBs. All these components operate integrally and indispensably to each other.

For eliminating the error propagation effects, enhancing the error detection capability, and maintaining the GOB/slice resynchronization, we have integrated these three algorithms into an error robust coding scheme for the H.263 standard-compatible video transmission. In practice, the overall scheme is especially suitable for the real time implementation due to its low memory requirement and computation complexity.

REFERENCES

1. R. Schaphorst, *Videoconferencing and Videotelephony: Technology and Standards*, Norwood, MA: Artech House, 1997.
2. T. J. Lynch, *Data Compression: Techniques and Applications*, Lifetime Learning Publications, 1985.
3. *Video Coding for Low Bit Rate Communication*, Draft ITU-T Recommendation H.263, May 1996.
4. H. Liu and M. El Zarki, "Transmission of H.263 video over wireless channels," in *Proc. SPIE Conf.*, vol. 2668, Jan. 1996.
5. U. Horn, B. Girod, and B. Belzer, "Scalable video coding with multiscale motion compensation and unequal error protection," in *Proc. Int. Symp. Multimedia Communications and Video Coding*, pp. 475-482, New York, NY, Oct. 1995.
6. S. Lin, D. J. Costello, and M. J. Miller, "Automatic repeat error control schemes," *IEEE Communications Magazine*, vol. 22, pp. 5-17, 1984.
7. S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," in *Proc. ICC '95*, pp. 1778-1783, Seattle, WA, Jun. 1995.
8. P. C. Chang and M. C. Chien, "Interleaving and error concealment for MPEG video over ATM networks," in *Proc. SPIE Conf. Electronic Imaging and Multimedia Systems*, vol. 2898, pp. 271-282, Beijing, China, Nov. 1996.
9. C. C. Liao, T. H. Lee, and P. C. Chang, "Error concealment with macro-block interleaving for H.263 video transmissions," in *Proc. IPPR Conf. Computer Vision, Graphics, and Image Processing, CVGIP '97*, pp. 570-577, Tai-Chung, Taiwan, Aug. 1997.
10. M. H. Willebeek-LeMair, Z. Y. Shae, and Y. C. Chang, "Robust H.263 video coding for transmission over the internet," in *Proc. INFOCOM '98*, pp. 225-232, San Francisco, CA, Mar. 1998.
11. *Video Coding - H.261, H.263*, ITU-T SG16 Q15, ftp://standard.pictel.com/video-site.
12. *Video Coding for Low Bit Rate Communication*, Draft ITU-T Recommendation H.263 Version 2, Jan. 1998.
13. T. H. Lee and P. C. Chang, "Error Robust H.263 Video Coding with Video Segment Regulation and Precise Error Tracking," *IEICE Trans. on Communications*, vol. E84-B, no. 2, pp. 317-324, Feb. 2001.
14. T. H. Wang, *Error Detection Using Data Embedding for H.263 Compatible Video Coding*, M.S. dissertation, Department of Electrical Engineering, National Central University, Jung-Li, Taiwan, 2000.
15. P. C. Chang and T. H. Lee, "Precise and Fast Error Tracking for Error-Resilient Transmission of H.263 Video," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 4, pp. 600-607, Jun. 2000.
16. J. Song and K. J. R. Liu, "A data embedding scheme for H.263 compatible video coding," in *Proc. ISCAS '99*, vol. 4, pp. 390-393, May 1999.
17. P. M. Crespo, R. M. Pelz, J. P. Cosmas, and J. Garcia-Frias, "Results of channel error profiles for DECT," *IEEE Trans. Communications,* vol. 44, pp. 913-917, Aug. 1996.
18. *Videoconferencing for Circuit Switched Networks (ISDN and POTS) - H.320, H.324*, ITU-T SG16 Q11, ftp://standard.pictel.com/lbc-site.