PAPER

Error Robust H.263 Video Coding with Video Segment Regulation and Precise Error Tracking*

Tien-Hsu LEE†, *Nonmember and* **Pao-Chi CHANG**†, *Regular Member*

SUMMARY This paper presents an error resilient H.263 video compression scheme over noisy channels. The start codes in the H.263 bit stream syntax, which inherently provide the resynchronization functionality for error handling, may cause significant error damage if they are incorrectly decoded. Therefore, we develop a video segment regulation algorithm at the decoder to efficiently identify and correct erroneous start codes and block addresses. In addition, the precise error tracking technique is used to further reduce the error propagation effects. After performing the video segment regulation, the decoder can report the exact addresses of detected corrupt blocks back to the encoder via a feedback channel. With these negative acknowledgments, the encoder can precisely calculate and trace the propagated errors by examining the backward motion dependency for each pixel in the current encoding frame. With this precise tracking strategy, the error propagation effects can be terminated completely by INTRA refreshing the affected blocks. Simulation results show that the proposed scheme yields significant video quality improvements over the motion compensated concealment by gains of 4.1 to 6.2 dB PSNRs at bit rate around 35 kbps in error-prone DECT environments. In particular, this scheme complies with the H.263 standard and has the advantages of low memory requirement and low computation complexity that are suitable for practical real-time implementation.
key words: *H.263, H.263+, video segment regulation, error tracking, error concealment*

1. Introduction

ITU-T H.263 [1] is a low bit rate video coding standard which has been successfully used for many applications, such as video conferencing and video telephony. Its version 2, known as H.263+ [2], was also ratified in January 1998 for improving compression performance, providing additional features, and supporting various networks more efficiently. By using motion estimation/compensation, discrete cosine transform (DCT), and variable length coding, significant compression gains can be achieved. However, some of these techniques are inherently very sensitive to the channel disturbances. A single error in the H.263 video stream may propagate in both the spatial and temporal domains, and causes serious quality degradation.

Due to the use of the variable length coding, the erroneous compressed data usually can not be decoded correctly until the next resynchronization point, i.e., the start of next group of blocks (GOB). Consequently, after the position the error occurred, all data in the following blocks of the same GOB are usually destroyed in the spatial domain. To provide enhanced error resilience capability, the H.263+ standard provides a new negotiable coding option mode, called the slice structured mode [3], [4], in which the original GOB layer in H.263 is replaced by a more flexible slice structure. All macroblocks (MBs) of one slice can be independently decoded because no data dependencies such as the prediction of motion vectors (MVs) can be allowed to cross the slice boundaries within the current picture. Therefore, a slice with shorter length can stop spatial error propagation more effectively.

In the temporal domain, since the I-picture and the P-picture are reference frames in the motion estimation/compensation, errors will propagate to all the following frames until all the erroneous MBs are refreshed by INTRA-mode coding. However, the encoder is not aware of the errors existing in the decoded bit stream in general and therefore becomes asynchronous with the decoder state. Thus, at the decoder a MB in the current decoding picture may be overwritten by the damaged area in the reference picture due to the motion compensation. Consequently, the reconstructed video quality is deteriorated and the error propagation continues.

Several techniques have been proposed to limit the effects of error propagation, such as the unequal error protection [5], the automatic retransmission request (ARQ) [6], and the error concealment [7]–[9]. The first two techniques, however, are required to modify the bit stream syntax and thus are incompatible with the standard. The error concealment techniques at the receiver have been proved to efficiently improve the image quality and reduce the error damage substantially. Most of the error concealment techniques utilize the information of neighbors in one or more domains to estimate the erroneous block. However, the success of the error concealment relies on the essential assumptions that the locations of erroneous blocks are identified accurately and all neighbors used for the concealment contain no errors. Unfortunately, these assumptions usually do not hold. Therefore, the performances of error concealment techniques degrade and they seldom terminate the temporal error propagation completely.

Manuscript received January 19, 2000.
Manuscript revised June 15, 2000.
†The authors are with the Department of Electrical Engineering, National Central University, Chung-Li, Taiwan 320, R.O.C.
*This work was supported by the National Science Council, Taiwan, under Grant NSC-88-2213-E-008-031.

For H.261, Chu and Leou presented an error detection and error concealment approach successful for combating transmission errors [10]. The constraints are that all GOBs within a picture can be correctly located. Many error concealment studies also avoid this problem by assuming correct decoding of headers. However, in H.263 headers, the start codes and the group numbers (or MB addresses) provide the information on the locations where to put the decoded bit stream and the basic resynchronization points for error handling. Consequently, an incorrectly decoded header may cause disastrous effects. Unequal error protection or ARQ may solve this problem. However, as we just mentioned, they are not H.263 syntax compatible. The IJG (Independent JPEG Group) provided a default resynchronization method for error recovery in still images [11], assuming the decoder is unable to back up. Besides, the restart marker has been studied to improve the error robustness [12], [13]. Those works mainly investigated on the positioning of markers, however, the marker error itself is not considered. Chang, Lee, and Hsu developed a resynchronization regulation algorithm for the progressive JPEG decoding [14]. It successfully regulates the restart interval order and effectively corrects most of the errors in restart markers. Based on this concept, we further develop a segment regulation algorithm to H.263 video coding for reducing such error effects.

A different approach to avoiding the error accumulation is INTRA coding the video with the penalty of rate increase. The usage of a feedback channel for error tracking and recovery after transmission errors has been adopted in Appendix I of H.263+ [2], [15]. In the error-tracking strategy, the encoder considers spatial-temporal error propagation caused by motion compensated prediction as well as the delay until the reception of the feedback message. To evaluate the feedback message, the encoder needs to continuously record the MV information during the encoding of each frame. Finally, the evaluated severely contaminated MBs are coded in INTRA mode for stopping temporal error propagation. Steinbach et al. proposed an error compensation strategy based on a feedback channel [16]. With the analysis of temporal dependencies of MBs in successive frames, this feedback approach leads to rapid quality recovery by reconstructing the error propagation effects at the encoder and selecting severely affected regions to be INTRA refreshed. In order to use the minimum number of INTRA MBs to terminate the error propagation, it is necessary to track errors more precisely. Consequently, we exploit the *pixel-based backward motion dependency* to perform error tracking in contrast to Steinbach's MB-based forward approach [17].

In this work, we propose an error robust H.263 coding with the video segment regulation (VSR) and the precise error tracking (PET) techniques to reduce the error damage from both the headers and video data. VSR reduces the damage from erroneous start codes and PET terminates the temporal error propagation of video signals. Additionally, VSR also provides more reliable feedback acknowledgements on the locations of erroneous MBs for performing PET. With both techniques applied, this system is much robust against errors occurred in all locations.

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 video segment regulation approach applied in the decoder. The precise error tracking algorithm used in the encoder is described in detail in Sect. 4. Subsequently, the simulation results of our algorithms are demonstrated in Sect. 5. Finally, Sect. 6 concludes our work.

2. System Architecture

The decoder and encoder schemes we propose are depicted in Fig. 1 and Fig. 2, respectively. 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 VSR technique in the decoder to identify and correct erroneous start codes and block addresses. A video segment can be a GOB in H.263 or a slice in H.263+. The decoder preprocesses the received com-

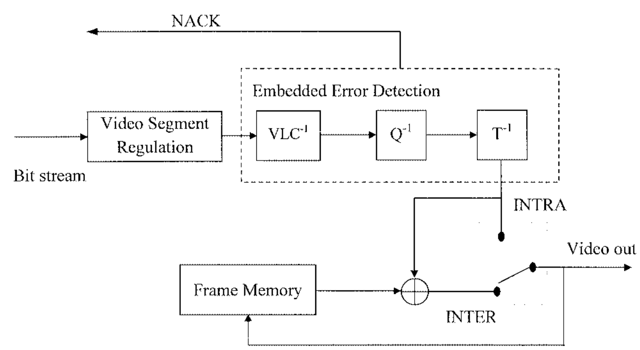

Fig. 1 Block diagram of H.263 decoder with video segment regulation.

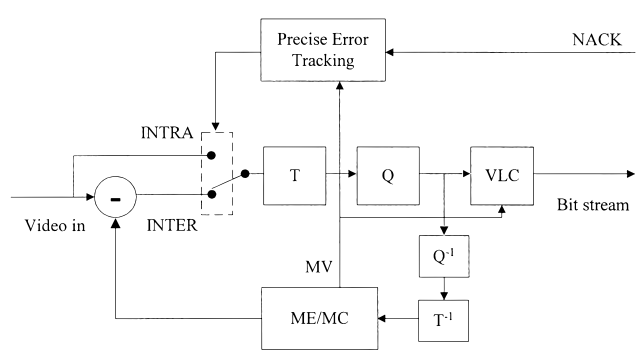

Fig. 2 Block diagram of H.263 encoder with precise error tracking.

pressed 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 (defined later in Sect. 3) 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 occurred in start code numbers can be corrected and consequently all GOB (or slice) locations are uniquely identified. Finally, the embedded error detection mechanism provided by the H.263 decoder [18], such as the validity of Huffman codes and the range of MVs, can discover the exact locations of erroneous decoded MBs. These negative acknowledgements (NACKs) can be sent back to the encoder for performing error tracking or to the decoder for performing 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 strategy, i.e., the PET, uses the pre-stored 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. To sum up, this algorithm is able to track the actual error propagation and completely terminate it. The earlier the feedback-transmission (NACK) arrives, the sooner the error-propagation stops.

3. Video Segment Regulation Algorithm

Due to the use of the variable length coding, 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 the 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 a 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 propagate. 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 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.

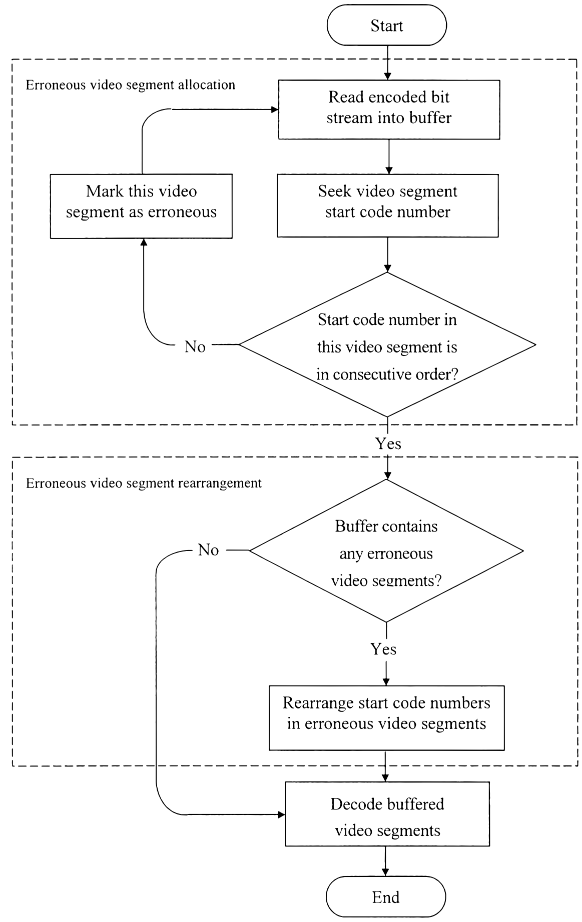

Fig. 3 Flow chart of video segment regulation at decoder.

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. 3. The proposed scheme 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 code and its corresponding encoded video segment. According to the GOB structure syntax shown in Fig. 4, 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 increment order. A start

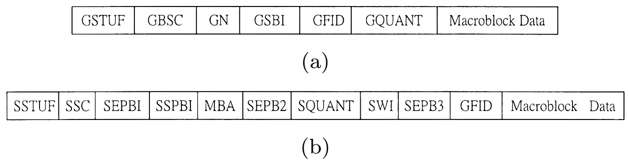

Fig. 4 Bit stream syntax diagram: (a) H.263 GOB layer and (b) H.263+ slice layer.

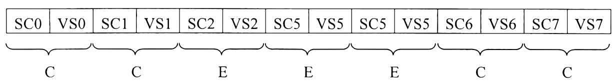

Fig. 5 Illustration of erroneous video segment allocation (C: correct, E: erroneous).

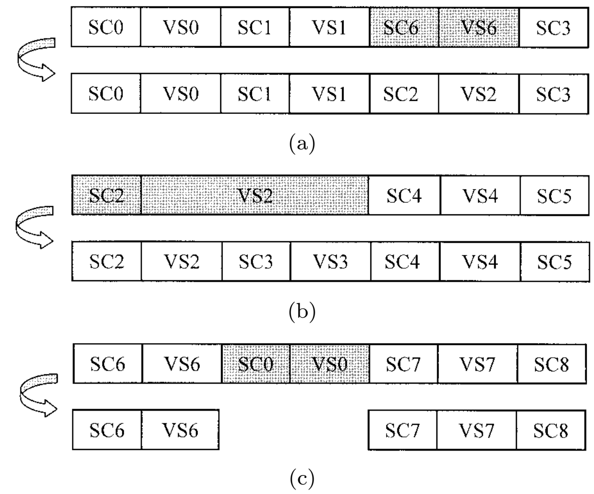

Fig. 6 Examples of erroneous video segment rearrangement: (a) forced start code assignment, (b) lost start code reconstruction, and (c) extra start code erasure.

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. The example in Fig. 5 illustrates this erroneous video segment allocation procedure. Three start codes (SC) and their corresponding video segments (VS) are labeled as erroneous (E), and the others are correct (C). It is particularly worth to note that SC2 + VS2 and the second SC5 + VS5 should be classified as erroneous since they are consecutive only in one side.

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.

(1) Forced start code assignment ($N_{\mathrm{found}} = N_{\mathrm{need}}$): All erroneous start codes are forcedly changed to the corresponding desired start code numbers that are in the consecutive order.

(2) Lost start code reconstruction ($N_{\mathrm{found}} < N_{\mathrm{need}}$): The lost start code is reconstructed by searching the erroneous video segments for the bit pattern which has the minimum Hamming distance to the desired one.

(3) Extra start code erasure ($N_{\mathrm{found}} > N_{\mathrm{need}}$): This case rarely happens in the H.263 as compared to the JPEG because the H.263 start code length is 17 bits long (0000 0000 0000 0000 1, without GN or MBA) while the JPEG restart marker is 8 bits (1111 1111) only. 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.

The search directions for the desired start codes can be forward, backward, or even both. In the case of both direction searches, the candidates with the minimum total Hamming distances will be chosen. Figure 6 offers examples of the erroneous start code rearrangement. In Fig. 6(a) ($N_{\mathrm{found}} = N_{\mathrm{need}} = 3$), the start code SC6 should be changed to SC2. Of course, the video segment 6 (VS6) will be decoded as VS2. In Fig. 6(b) ($N_{\mathrm{found}} = 2$, $N_{\mathrm{need}} = 3$), the lost start code SC3 is searched within original VS2 by comparing the Hamming distances of all bit patterns to SC3. Then, the bit pattern with the minimum distance is forcedly changed to SC3 and the subsequent data is recognized as VS3. In Fig. 6(c) ($N_{\mathrm{found}} = 3$, $N_{\mathrm{need}} = 2$), the start code SC0 and its VS0 are erased 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 can not 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.

4. Precise Error Tracking Algorithm

Figure 2 has shown the coding scheme of the H.263 encoder with the precise error tracking control. NACKs with the information on unsuccessfully decoded image blocks are sent back to the encoder via a feedback channel. Once a NACK is received, the encoder performs error tracking to determine if the current encoding MB is contaminated by the erroneous MBs in the past frames. If this happens, this MB becomes a candidate for INTRA-mode refreshing.

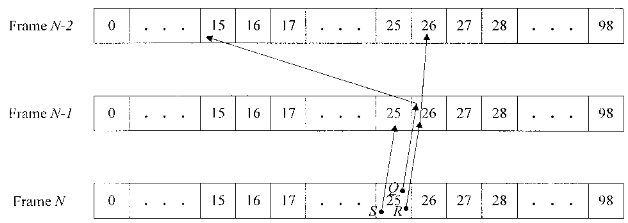

Fig. 7 Illustration of pixel-based backward motion dependency tracking at encoder.

Table 1 PSNR (dB) comparisons of various slice sizes (number of MBs) at different bit error rates for MC.

Bit error rate \ Slice size	5	8	11
no error	34.18	34.18	34.18
1.0E-4	30.73	30.61	30.22
2.0E-4	28.80	28.67	28.04
1.0E-3	24.41	24.25	23.68

The MV of the MB produced in the motion estimation progress indeed provides adequate information for accurately tracing error propagation at the encoder. The example in Fig. 7 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 $\mathrm{N}-1$. MB 26 of frame $\mathrm{N}-1$ refers to MBs 15, 16, 26, and 27 in frame $\mathrm{N}-2$. MB 25 of frame $\mathrm{N}-1$ is coded in INTRA mode. If the encoder receives a NACK, which indicates an error occurred at MB 15 of frame $\mathrm{N}-2$, 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 $\mathrm{N}-2$. As shown in Fig. 7, 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. 7 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 $\mathrm{N}-1$. In such a case, pixel S in Fig. 7 is certainly claimed to be clean.

With this pixel-based error tracking strategy, the degree of damage caused by the error propagation can be calculated. Every MB in frame N is scored by the contamination ratio (CR) defined as

$$CR = \frac{N_C}{N_B} \tag{1}$$

where N_C is the number of contaminated pixels and N_B is the total number of pixels in a MB. The contaminated MBs, i.e., MBs with $\mathrm{CR} > 0$, should be INTRA refreshed for complete termination of error propagation to succeeding frames.

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 with the largest CR values to perform INTRA refreshing. Of course, in such cases the error propagation may not be stopped completely.

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.

5. Simulation Results

In our simulations, the following notations are adopted:

- Error free (EF);
- Motion compensated concealment (MC);
- Video segment regulation (VSR);
- Precise error tracking with the contamination ratio larger than K (PET ($\mathrm{CR} > K$));
- Precise error tracking with M MBs of the largest contamination ratios, i.e., the number of INTRA-mode refreshed MBs with the largest contamination ratios is at most M (PET ($\mathrm{CRN} = M$));
- Precise error tracking with video segment regulation (VSR + PET).

Three video sequences, *Susie*, *Foreman*, and *Salesman*, are tested in the simulations. They are QCIF (176×144) format, reference frame rate at 30 Hz, and 150, 400, 400 frames in length, respectively. These sequences are compressed by the H.263 without four negotiable options and the rate control. The encoded frame rate is kept to be 10 Hz. By default, MC is used for the error recovery of detected erroneous MBs at the decoder. Here, 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 100 simulation iterations for different offsets in the bit error patterns.

At first, we demonstrate the effects of the length of the slice, i.e., the length of the minimum resynchronization unit. Uniformly distributed random errors are added into the encoded bit stream. From the simulation results of sequence *Susie* shown in Table 1, we can find that the longer the slice length is used, the severer the

Table 2 PSNR (dB) comparisons of various slice sizes (number of MBs) at different bit error rates for VSR.

Bit error rate \ Slice size	5	8	11
no error	34.18	34.18	34.18
1.0E-4	31.20	30.98	30.39
2.0E-4	29.50	29.03	28.43
1.0E-3	25.12	24.71	24.12

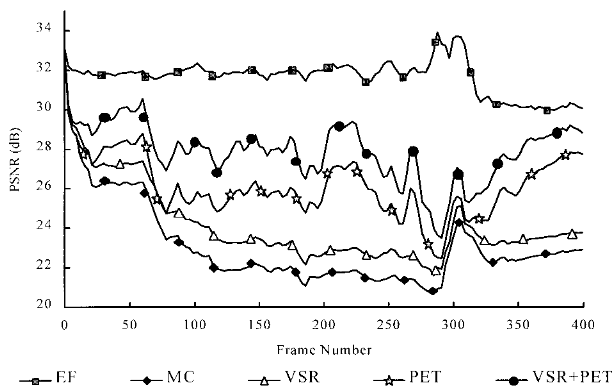

Fig. 8 PSNR simulation results for sequence *Foreman* in DECT1 channel.

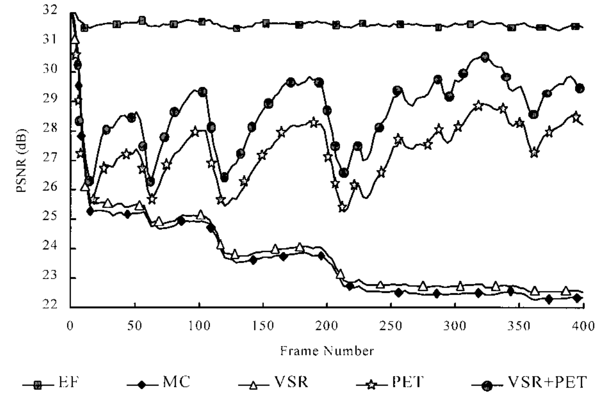

Fig. 9 PSNR simulation results for sequence *Salesman* in DECT2 channel.

video quality degrades. In the meantime, the results of the decoder additionally performing the VSR strategy are also shown in Table 2. VSR indeed improves the average video quality, especially in the condition of high error rate and short slice size (i.e., frequent use of start codes). The performance of the error concealment is also improved due to the decreased number of incorrectly detected locations of erroneous blocks. Moreover, errors may erase or produce fake picture start codes (PSCs) in general. Accordingly, the number of decoded frames will be incorrect and the playback time may not be accurate. VSR can drastically reduce this error probability and mostly resolve this problem.

In the following experiments, video sequences are tested in two simulated channels DECT1 and DECT2, which are generated by a correlated Rayleigh fading model for the DECT system at 14-km/h speed [19]. DECT1 contains a bit error pattern at $Es/No = 20$ dB (the equivalent bit error rate of 0.264%). DECT2 contains a bit error pattern at $Es/No = 12$ dB (the equivalent bit error rate of 1.51%). We will use the reasonable fixed slice size of five for the better error-resilient capability. For smooth human speech conversations, the ITU-T Recommendation G.114 specifies a maximum round-trip delay of 400 ms [20]. Here, the round-trip delay for receiving a NACK is assumed 300 ms.

Figure 8 first shows the performance of the proposed scheme for sequence *Foreman* in DECT1 channel. Here, the maximum number of INTRA-refreshed MBs is limited to eleven per frame for maintaining a constant bit rate. Observed from Fig. 8, VSR increases the PSNR values by more than 1.05 dB compared with MC only. PET can further improve 2.03 dB in average while the bit rate increases 14% due to INTRA coding. It is particularly worthy of note that VSR + PET obtains additional 1.8 dB gains over PET only while keeping even lower bit rate. Rapid error recovery is achieved by PET and the video quality is consistently maintained. VSR can correct most errors occurred in start codes and gain additional improvements in the video quality and the bit rate. Figure 9 also shows the performance of the proposed scheme for sequence *Salesman* in DECT2 channel. Similar conclusions can be made. The video quality processed by MC only deteriorates seriously and can not be recovered until the end of sequence in such a high error rate environment. With the proposed coding scheme, the error propagation can be rapidly (even fully if PET (CR $>$ 0) is used) terminated and the damage can be concealed to give the most satisfactory performance.

It is especially interesting to estimate the effectiveness of VSR, which uses the Hamming distance as the distortion measure. Due to burst errors in mobile channels, the lost start code reconstruction is the most frequent occurrence in three cases of the erroneous video segment rearrangement. Table 3 shows the simulation results for the lost start code reconstruction only. We can find that the Hamming distance may possibly result in wrong detection of start codes. However, such a wrong decision usually generates invalidity in the adjacent video segments, which could be discovered by the embedded error detection and compensated by the error concealment. From the simulation results, it still can recover 60–70% lost start codes and improve the average video PSNR by more than 0.5 dB over MC only in burst-error mobile channels. Consequently, in most situations wrong start code detection will not cause severer degradation than simple discard of the wrong GOB or slice.

Table 3 Simulation results of the lost start code reconstruction only for VSR.

Channel	DECT1		DECT2	
Sequence	*Susie*	*Foreman*	*Susie*	*Foreman*
Average PSNR increase over MC (dB)	0.46	0.78	0.56	0.89
Wrong start code detection probability (%)	35.6	29.2	38.4	34.3

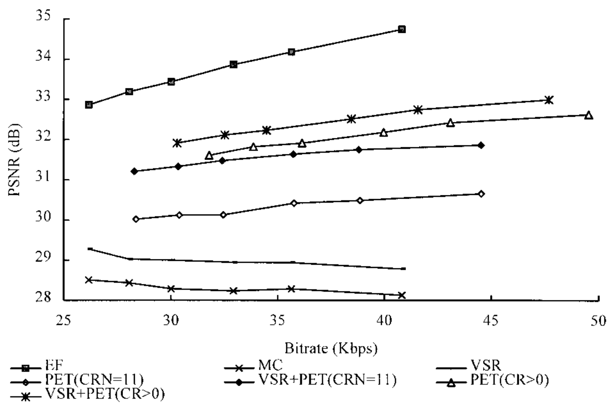

Fig. 10 Performance comparisons of different methods for sequence *Susie* in DECT1 channel.

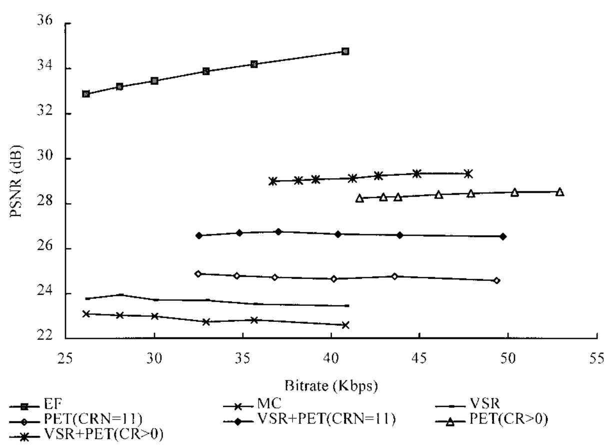

Fig. 11 Performance comparisons of different methods for sequence *Susie* in DECT2 channel.

The average rate-distortion relationships for sequence *Susie* transmitted in DECT1 and DECT2 channels are shown in Fig. 10 and Fig. 11, respectively. As we know, INTRA-mode coding usually results in the overshoot of the bit rate. The purpose for these experiments is to compare the performances of different methods at the same video bit rate and the same error rate. In both cases, VSR + PET (CR $>$ 0) has shown the best performance among these methods operating at the same bit rate. More specifically, VSR + PET (CR $>$ 0) improves the video quality by 4.1 to 6.2 dB over MC only at the bit rate around 35 kbps. Therefore, both VSR and PET can provide a robust coding scheme in error-prone environments. A noticeable phenomenon can be also found in these figures that sometimes the PSNR decreases with the increase of encoded bit rate, especially for the case of high bit error rate (DECT2). This is because the fine quantization results in more encoded bits, which are vulnerable to the error damage. We conclude that in error-prone environments, fine quantization without a good error recovery strategy does not improve the video quality at all. In other words, the bit rate budget should be spent on INTRA coding for the most desirable MBs instead of fine quantization.

6. Conclusions

For eliminating the error propagation effects, the error concealment needs accurate corrupted locations and the INTRA refreshing method needs accurate error tracking. We have presented an integrated video segment regulation and precise error tracking coding scheme for the error robust H.263 video transmission. The video segment regulation pre-processes the bit stream at the decoder to efficiently identify and correct erroneous block addresses. This procedure is also helpful to sending back correct locations of damaged blocks for performing the precise error tracking at the encoder. Besides, by utilizing the MVs generated in the regular encoding process and tracing the backward motion dependency for each pixel, contaminated MBs can be accurately tracked and the error propagation can be completely terminated by INTRA coding the affected MBs. This approach achieves rapid error recovery and complies with the standard. In practice, the video segment regulation and the precise error tracking techniques are especially suitable for real-time implementation due to their low memory requirement and computation complexity.

References

[1] Draft ITU-T Recommendation H.263, Video Coding for Low Bit Rate Communication, May 1996.

[2] Draft ITU-T Recommendation H.263 Version 2, Video Coding for Low Bit Rate Communication, Jan. 1998.

[3] G. Cote, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video coding at low bit rates," IEEE Trans. Circuits & Syst. for Video Technology, vol.8, no.7, pp.849–866, Nov. 1998.

[4] S. Wenger, G. Knorr, J. Ott, and F. Kossentini, "Error resilience support in H.263+," IEEE Trans. Circuits & Syst. for Video Technology, vol.8, no.7, pp.867–877, Nov. 1998.

[5] U. Horn, B. Girod, and B. Belzer, "Scalable video coding with multiscale motion compensation and unequal error protection," Proc. Int. Symp. Multimedia Communications and Video Coding, pp.475–482, New York, Oct. 1995.
[6] S. Lin, D.J. Costello, and M.J. Miller, "Automatic repeat error control schemes," IEEE Commun. Mag., vol.22, pp.5–17, 1984.
[7] S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," Proc. IEEE ICC '95, pp.1778–1783, Seattle, June 1995.
[8] Y. Wang and Q.F. Zhu, "Error control and concealment for video communication: A review," Proc. IEEE, vol.86, no.5, pp.974–997, May 1998.
[9] N.P. Galatsanos and A.K. Katsaggelos, eds., Error Concealment in Encoded Video Streams, Kluwer Academic Publishers, 1998.
[10] W.J. Chu and J.J. Leou, "Detection and concealment of transmission errors in H.261 images," IEEE Trans. Circuits & Syst. for Video Technology, vol.8, no.1, pp.74–84, Feb. 1998.
[11] Independent JPEG Group, JPEG Software Release 6A, Feb. 1996.
[12] K.Y. Yoo, "Adaptive resynchronisation marker positioning method for error resilient video transmission," IEE Electronics Letters, vol.34, no.22, pp.2084–2085, Oct. 1998.
[13] G. Cote, S. Shirani, and F. Kossentini, "Robust H.263 video communication over mobile channels," Proc. IEEE ICIP '99, vol.2, pp.535–539, Kobe, Japan, Oct. 1999.
[14] T.H. Lee, H.H. Hsu, and P.C. Chang, "Robust transmission of progressive JPEG image coding in wireless communications," Proc. IEEE ICCE '99, pp.234–235, Los Angeles, June 1999.
[15] N. Farber, B. Girod, and J. Villasenor, "Extensions of ITU-T recommendation H.324 for error-resilient video transmission," IEEE Commun. Mag., vol.36, no.6, pp.120–128, June 1998.
[16] E. Steinbach, N. Farber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," IEEE Trans. Circuits & Syst. for Video Technology, vol.7, no.6, pp.872–881, Dec. 1997.
[17] P.C. Chang and T.H. Lee, "Precise and fast error tracking for error-resilient transmission of H.263 video," IEEE Trans. Circuits & Syst. for Video Technology, vol.10, no.4, pp.600–607, June 2000.
[18] Signal Processing and Multimedia Lab., Univ. British Columbia, TMN 8 (H.263+) encoder/decoder, Version 3.2.0, Aug. 1998.
[19] G. Soeder, Modelierung, Simulation und Optimierung von Nachrichtentechnik-Systemen, Springer, 1993. (in German)
[20] ITU-T Recommendation G.114, One-way transmission time, Feb. 1996.

Tien-Hsu Lee was born in Hsinchu, Taiwan, in 1971. He received the B.S. degree in 1993 from National Cheng Kung University, Taiwan and the Ph.D. degree in 2000 from National Central University, Taiwan, both in electrical engineering. His current interests are in the area of image/video coding, error prevention and concealment techniques, and wireless network applications.

Pao-Chi Chang was born in Taipei, Taiwan, in 1955. He received the B.S. and M.S. degrees from National Chiao Tung University, Taiwan, in 1977 and 1979, respectively, and the Ph.D. degree from Stanford University, California, 1986, all in electrical engineering. From 1986–1993, he was a research staff member of the department of communications at IBM T.J. Watson Research Center, Hawthorne, New York. At Watson, his work centered on high speed switching system, efficient network design algorithms, network management, and multimedia teleconferencing. In 1993, he joined the faculty of National Central University, Taiwan. His current interests are in the area of speech/image coding, video conferencing, and video coding over high speed networks and wireless communications.