Precise and Fast Error Tracking for Error-Resilient Transmission of H.263 Video

Pao-Chi Chang and Tien-Hsu Lee

Abstract-In this letter, a precise error-tracking scheme for robust transmission of real-time H.263 video is presented. By utilizing a feedback channel, the decoder reports the addresses of corrupted blocks induced by transmission errors back to the encoder. With these negative acknowledgments, the encoder can precisely calculate and track the propagated errors by examining the backward motion dependency for each pixel in the current encoding frame. With this precise tracking, the error-propagation effects can be terminated completely by INTRA refreshing the affected macroblocks. In addition, by utilizing the four-corner tracking approximation and the linear motion model, a fast algorithm is also developed to further reduce the computation and memory requirements. The simulations show that both schemes yield significant video quality improvements in error-prone environments. The advantages of the low memory requirement and the low computation complexity are particularly suitable for real-time implementation.

Index Terms—Error concealment, error propagation, error tracking, H.263, video coding.

I. INTRODUCTION

T HE ITU-T H.263 [1] is a low bit-rate video coding standard that has been used for video conferencing over various networks. Significant compression gains can be achieved by using motion estimation/compensation, discrete cosine transform (DCT), and variable length coding (VLC). However, some of these techniques are inherently extremely sensitive to the channel disturbances. A single error in the H.263 video stream may propagate in both spatial and temporal domains and consequently cause serious quality degradation.

Due to the use of VLC, the erroneous compressed data usually cannot be correctly decoded until the next resynchronization point, i.e., when the start of the following group of blocks (GOB) appears. Consequently, all data in the following blocks of the same GOB are usually destroyed in the spatial domain. Moreover, because of the out-of-synchronization between the encoder state and the decoder state, the errors may also propagate in the temporal domain. In general, errors existing in the decoded bit stream are hardly perceived by the encoder. Thus, at the decoder a macroblock (MB) may be overwritten by the damaged area pointed by the received motion vector (MV). Consequently, the reconstructed video quality is deteriorated and the error-propagation continues.

Several techniques, such as unequal error protection [2], ARQ [3], and error concealment [4]–[6], have been proposed to limit the effects of error propagation. The first two techniques, however, require bitstream syntax modifications and are not com-

The authors are with the Department of Electrical Engineering, National Central University, Chung-Li, Taiwan 32045 (e-mail: pcchang@ee.ncu.edu.tw).

Publisher Item Identifier S 1051-8215(00)04897-7.

patible with the standard. The error-concealment techniques at the receiver have been proved their ability to effectively reduce the error damage. Most of the error-concealment techniques utilize the information from neighbors in one or more domains to estimate the erroneous blocks. Unfortunately, although these concealment techniques can substantially improve the video quality, they hardly avoid or even terminate the temporal error propagation completely.

A different approach to avoiding error accumulation is to INTRA code the video with the penalty of rate increase. The H.263 standard requires that each MB in P-pictures shall be coded in INTRA mode at least once for each 132 frames. A sequential INTRA update generates a smooth bandwidth profile, but it takes long time to recover from errors. The Intelligent Macroblock Update (IMU) technique, proposed by Willebeek-LeMair *et al.*, is used to analyze the temporal dependencies of MB's in successive frames and selectively update the MB's that have the extreme impact on later frames [7]. This technique can improve the performance, but it still cannot terminate the error propagation completely.

The usage of a feedback channel for error robust video transmission has been adopted by the mobile extension of H.263. Steinbach *et al.* proposed an error-compensation strategy based on a feedback channel [8]. With the analysis of temporal dependencies of MB's in successive frames, this feedback approach leads to rapid quality recovery by reconstructing the error propagation effects at the encoder and by selecting seriously affected regions to be INTRA refreshed. However, in such an algorithm, the assumption that errors distribute uniformly within each MB results in over-estimation of error-propagated areas. Besides, the forward dependency trees may also span too widely and grow to unnecessarily large sizes, especially for those MB's with large MV's and the networks with long round-trip delays.

In order to use the minimum number of INTRA MB's to terminate the error propagation, it is necessary to track errors precisely. A pixel-based postprocessing of late asynchronoous transfer mode (ATM) cells in packet video can be found in [9]. The late cells received from the auxiliary buffer are processed and properly added to the current decoded picture for preventing accumulation of image artifacts. In contrast to solving late cell problem, the pixel-based backward motion dependency is utilized in our work to perform error tracking for erroneous MB's in error-prone environment. In particular, the proposed strategy uses the pre-stored MV's only and traces the motion dependency for each pixel backward to the unsuccessfully decoded frame. The encoder can thus exactly evaluate how seriously each MB in the current encoding frame is affected. At last, all or parts of the contaminated MB's can be selected to refresh by INTRA-mode coding. It is important to note that this algorithm is able to track the actual error propagation and effectively terminate it. In fact, precise error recovery can also be

Manuscript received January 19, 1999; revised October 27, 1999. This work was supported by the National Science Council under Grant NSC 87-2213-E-008-036, Taiwan, R.O.C. This paper was recommended by Associate Editor Y. Wang.



Fig. 1. Block diagram of H.263 encoder with error tracking.



Fig. 2. Illustration of pixel-based backward motion dependency tracking.

achieved by re-encoding the last frames after a feedback acknowledgment arrives. However, the computational complexity is much higher. In addition, we also develop a fast algorithm to further reduce the computation complexity and to decrease the memory requirement by utilizing the approximation of MB's corner tracking and the assumption of constant-velocity motion model.

In this letter, the detail of the precise error tracking (PET) algorithm is described in Section II, a fast algorithm of the proposed method is presented in Section III, the simulation results of the proposed algorithms are given in Section IV, and at last, the research is concluded in Section V.

II. PRECISE ERROR-TRACKING ALGORITHM

The coding scheme of the H.263 encoder with the error tracking control is shown in Fig. 1. Negative acknowledgments (NACK's) 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 whether the current encoding MB is contaminated by the erroneous MB's in the past frames. If this happens, this MB becomes a candidate for INTRA-refresh coding.

The dependencies of MB's in successive frames are essential to the error tracking. The MV of the MB produced in the motion estimation indeed provides adequate information for accurately tracing error propagation in the encoder. Fig. 2 illustrates how to execute the PET for pixels Q, R, and S by employing the backward motion dependencies for a QCIF format video with 99 MB's. Any pixel's motion dependency can be found by tracing back the MV of the MB it belongs to. Assuming that, in Fig. 2, the prediction of MB 25 in frame N is obtained from MB 25 and 26 in frame N-1. MB 26 of frame N-1 refers to MB 15, 16, 26, and 27 in frame N-2. The encoder receives a NACK that indicates an error occurred at MB 15 of frame N-2 while working on frame N. Thus, we can backward trace every single pixel in MB 25 of frame N along the corresponding paths, i.e., the corresponding MV's, to see if it refers to the erroneous area, i.e., MB 15 of frame N-2. Based upon the illustration, pixel Q in Fig. 2 is then determined to be a contaminated pixel while pixel R is not. Likewise, the backward motion dependency structure for each pixel is built as in Fig. 2 and the error tracking procedure is performed for all pixels in frame N.

The backward motion dependencies, i.e., the paths shown in Fig. 2, can be mathematically designated as follows. The location of any pixel in an image frame is interpreted as a vector, denoted by \mathbf{P} . With the acknowledgment of the MV of pixel \mathbf{P} , denoted by $\mathbf{M}_{\mathbf{P}}$, the backward motion dependency of the adjacent frames for pixel \mathbf{P} is expressed by

$$\mathbf{P}^{(i-1)} = F^{(i)}(\mathbf{P}) = \mathbf{P}^{(i)} + \mathbf{M}_{\mathbf{P}}^{(i)}$$
(1)

where *i* is the frame number. $F(\cdot)$ can be interpreted as a function to map a pixel into the original location in the preceding frame. In general, the backward motion dependency of the *L*th preceding frame for pixel **P** in frame *N* is thus recursively expressed as

$$\mathbf{P}^{(N-L)} = F^{(N-L+1)} F^{(N-L+2)} \cdots F^{(N)}(\mathbf{P}).$$
 (2)

It is particularly important to note that there exists no MV for INTRA MB's, such as MB 25 of frame N-1 in Fig. 2. In such a case, $\mathbf{M}_{\mathbf{P}}$ should be alternatively substituted by infinity (∞) in the tracking procedure. Consequently, this pixel is definitely claimed to be clean, such as pixel S in Fig. 2. After all pixels in frame N are backward tracked with (2), all $\mathbf{P}^{(N-L)}$ are examined to learn if they fall into unsuccessfully decoded areas, denoted by D.

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

$$CR = \frac{N_C}{N_B} \tag{3}$$

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

In summary, the PET procedures for each encoding frame can be listed as follows:

For
$$(i = 0; i < \text{number of MB's in frame } N; i++)$$
{
 $CR_i = 0$
For $(j = 0; j < N_B; j++)$ {
For $(k = 1; k <= L; k++)$ {
Compute $\mathbf{P}_j^{(N-k)}$
}
If $(\mathbf{P}_j^{(N-L)} \in D)$ then $CR_i + = 1/N_B$
}
If $(CR_i > 0)$ then INTRA refresh the *i*th MB
}

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 and shifts 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.

Although INTRA MB terminates temporal error propagation, it usually generates a higher bit rate. If the rate increased by INTRA coding is higher than what we can afford, we need to limit the number of INTRA MB's in a frame. Several methods may be applied. For instance, we can select only a fixed number of MB's with the largest CR values. A simpler method is to limit the number of INTRA MB's by comparing the CR with a threshold T ($0 \le T < 1$). Only the MB's with CR > T will be INTRA coded. Of course, in such cases the error propagation may not be stopped completely.

The comparison between the pixel-based backward approach proposed above and Steinbach's MB-based forward approach can be illustrated by the example shown in Fig. 3. For simplicity, considering all MB's have the same MV(4, 4), we assume that MB 37 of frame N is contaminated by an error, marked as the shaded area in Fig. 3. In Steinbach's algorithm, the INTRA refresh candidates are determined by the forward dependencies, error weights (areas), and error energy. In addition, the error distribution inside each MB is also assumed uniform after each iteration of error energy calculation [8]. Consequently, MB 37 of frame N will propagate 3/16 error energy into MB 38 and 48, respectively, and 1/16 error energy into MB 49 of frame N+1by forward MB-dependency tracking. Therefore, four MB's, marked as the shaded area, are determined to be erroneous. Because of the assumption of uniform error distribution that diffuses the error into the whole MB, nine MB's are determined to be contaminated in frame N+2, and the number of contaminated MB's increases in a square manner as time goes by. Certainly, if only one pixel (or a small number of pixels) of a MB is affected, the error weight for this propagation path is very small and the corresponding MB in the current frame will not be INTRA refreshed under a rate constraint in practice. Oppositely, while using the proposed backward pixel-based PET algorithm, only four MB's, i.e., MB 37, 38, 48, and 49, in frames N+1 and N+2 are judged as corrupt. Moreover, in addition to storing MV's, the forward tracking algorithm calculates and stores summed absolute difference (SAD) values and transition weights to determine the order of INTRA refreshing for the contaminated MB's. The backward tracking algorithm only needs to store MV's and execute a few simple operations such as additions and modulo-16 divisions. Compared with the forward algorithm, the PET can backward trace exact propagated errors and require less computation complexity and memory requirement.

III. FAST ERROR-TRACKING ALGORITHM

Compared with the video coder itself, the PET algorithm is already relatively simple. Nevertheless, the fast algorithm that further reduces the complexity can still be developed. There are mainly two reasons for developing the fast error tracking (FET) algorithm. One is that if the end-to-end delay is relatively long so that it is larger than the assumed maximum time, we will lose the required MV information. The other is that if a NACK indicates that only one MB in a preceding frame is corrupted, it is unnecessary and time wasting to perform error tracking for each pixel in the current encoding frame, especially for the video with large-size format. Therefore, we simplify the PET by two approximations.

First, it is unlikely that all four corners refer to error-free (EF) area, while the inner pixels in a MB refer to contaminated area. Hence, backward motion dependencies are examined for only four corners of a MB, instead of each pixel, to speed up the tracking process. This procedure will dramatically reduce the computation complexity by 64 folds without the loss of much accuracy. The second approximation is based on the assumption of constant velocity motion. We utilize the linear motion prediction model for the FET. As illustrated in Fig. 4, the encoder receives the feedback NACK while working on frame N. The NACK informs that an error occurred in the *L*th preceding frame. Thus, we can approximately estimate the movement of the pixel **P** under backward error tracking by the first-order linear prediction as

$$\tilde{\mathbf{P}}^{(N-L)} = \mathbf{P}^{(N)} + L \times \mathbf{M}_{\mathbf{P}}^{(N)}.$$
(4)



Fig. 3. Example for illustrating estimated contamination area by different error-tracking methods. (a) Forward MB-based error tracking. (b) Backward pixel-based error tracking.



Fig. 4. Linear motion model for FET (L = 3).

From (4), it is obvious that we do not need to store the MV's of past frames at all henceforth save more memory.

These two approximations are not mutually exclusive but can be used alone or together. If any corner of an MB points to the erroneous area, this MB may perform either the PET or the FET using linear motion prediction to evaluate the CR value. Briefly speaking, this fast algorithm has the advantages of no extra memory requirement for storing MV's and a very low computation complexity.

IV. SIMULATION RESULTS

In our simulations, the following scenarios are compared:

- 1) EF;
- 2) Error concealment with temporal replacement (TR);
- 3) Motion-compensated concealment (MC);
- 4) PET with contamination ratio larger than K (PET (CR > K));

- 5) FET with four-corner approximation and contamination ratio larger than K (FET1 (CR > K));
- 6) FET with four-corner and linear motion prediction approximations and contamination ratio larger than K (FET2 (CR > K)).

At first, the video sequence Suzie is tested in the simulation. It is QCIF (176×144) format, reference frame rate at 30 Hz, and 150 frames in length. This sequence is compressed by H.263 without four negotiable options [1] and rate control. The encoded frame rate is kept to be 10 Hz and the obtained average bit rate is 31.88 kb/s. To make the effect of error damage and error concealment more observable, we add two errors located at the beginnings of two consecutive GOB's of frame 51, as shown in Fig. 5(b). The Suzie sequence consists of both fast and slow motion parts. Fast motion with shaking head appears within frame 40 to 90. Two kinds of error concealment techniques are used here for the comparison. One is the simple TR method, in which the two GOB's are replaced by the corresponding 22 MB's at the same location from the preceding frame. The other one is the MC. The motion compensation is carried out for each erroneous MB by using the average MV of upper and left neighboring MB's.

In general, the TR technique results in good quality only for slow-motion video. On the other hand, the MC performs much better even for fast-motion video. The images with the TR and MC are shown in Fig. 5(c) and (d), respectively. We observe that both methods result in poor image quality and errors propagate to succeeding frames in both spatial and temporal domains for a long period. In fact, the damage is still significantly perceptible and cannot be recovered at the end of the sequence.

Then, we assume the encoder receives the feedback NACK's that indicate the error locations in frame 51 while encoding frame 60. Note that the error tracking starts after the NACK was received and that the frames before cannot be recovered by error tracking, i.e., only error concealment. The simulation results for precise error tracking and fast error tracking are exhibited in Fig. 6. In the case of PET (CR > 0), all 33 contaminated MB's are INTRA refreshed and the error propagation to the succeeding frames is terminated immediately. Thus, rapid error recovery is achieved and the video quality is consistently maintained. The image quality of FET1 (CR > 0) is almost the same as the PET case, while only two among 33 contaminated MB's are missing. Therefore, 31 MB's need to perform all-pixel error tracking while 68 MB's only execute four-corner error tracking. The computation complexity, defined as the number of pixels needed to perform error tracking, reduces to (31 * 256 +68 * 4)/176 * 144 = 32% of the precise version in this case. In the case of FET2 (CR > 0), six contaminated MB's among 33 MB's are missing and two EF MB's are misjudged as being contaminated by errors. Some errors still exist in frame 60 and continuously propagate to frame 63. Nevertheless, the computation complexity reduces to $(29 \times 256 + 70 \times 4)/176 \times 144 = 30\%$ of the precise version in this case, and the memory to store MV's is no longer required.

We also depict the variation of the peak signal-to-noise ratio (PSNR) in Fig. 7(a). Although MC outperforms TR by 3–4.6 dB, it still suffers serious error-propagation effects. On the other hand, PET (CR > 0) yields significant SNR improvements



Fig. 5. Frame 51 of sequence *Suzie*. (a) EF. (b) Two erroneous GOB's. (c) TR of two GOB's. (d) MC of two GOB's.



Fig. 6. Video quality in the condition that two erroneous GOB's are concealed in frame 51 and NACK's are received on encoding frame 60. (a), (b) PET (CR > 0). (c), (d) FET1 (CR > 0). (e), (f) FET2 (CR > 0).

over MC by 5–1.4 dB in the following frames and exhibits almost the same quality as in the EF case. Note that the quality of PET is sometimes even better than EF because of a larger number of INTRA-coded MB's. With this technique, the error propagation can be terminated after the NACK is received and



Fig. 7. Performance of the proposed error-tracking algorithms for sequence *Suzie* in the condition of two erroneous GOB's. (a) PSNR. (b) Bit rate.



Fig. 8. PSNR results of the proposed error-tracking algorithms for sequence *Suzie* in DECT1 channel.

the damage can be concealed to give the most satisfactory performance. It is noteworthy that FET1 performs almost as well as PET and FET2 only degrades the image quality by about 0.5 dB. The overhead of the bit rate is only the overshoot in the INTRA refreshed frame 60, as illustrated in Fig. 7(b). However, this can be overcome by utilizing some strategies to limit the number of INTRA-refreshed MB's. We will demonstrate the results at the end of this section.

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. DECT1 contains a bit error



Fig. 9. PSNR results of the proposed error-tracking algorithms for sequence *Salesman* in DECT1 channel.



Fig. 10. PSNR results of the proposed error-tracking algorithms for sequence *Foreman* in DECT1 channel.

pattern at $E_s/N_o = 20$ dB (the equivalent bit error rate of 0.264%). DECT2 contains a bit-error pattern at $E_s/N_o = 12$ dB (the equivalent bit error rate of 1.51%). All sequences are compressed by H.263 without four negotiable options and rate control. The encoded frame rate is kept to 10 Hz and the obtained average bit rate is 28 kb/s. Fifty simulation iterations are averaged for different offsets in the bit-error pattern.

Fig. 8 first shows the PSNR performance for sequence Suzie in DECT1 channel. It is assumed that MC is performed on each misdecoded MB and the round-trip delay for receiving a NACK is 300 ms. We find that both PET and FET can track errors successfully and yield image quality as good as EF. Referring to Fig. 8, it is worth to note that the drastic drop in PSNR from frames 40-90 of PET and FET actually results from the unsuccessful MC in the fast-motion part of video and the error propagation caused by long round-trip delay, but not the inaccuracy of error tracking. Nevertheless, the PSNR recovers to near the EF performance once the high motion activity stops. The results for 400 frames of the sequence Salesman and Foreman are also shown in Figs. 9 and 10, respectively. The sequence Salesman exhibits relatively low motion while Foreman contains high motion activity including camera movement. Both PET and FET still show excellent error-tracking capability. The image quality degradation of PET and FET for Salesman is at most 1.6 dB,



Fig. 11. Robustness of the proposed error-tracking algorithms with different amount of round-trip delay (ms) for sequence *Suzie* in DECT1 channel.



____ EF ___ PET (CR>0) ____ FET2 (CR>0) ____ PET (CR>0.5) ___ PET (CRN=11) ___ MC

Fig. 12. Comparison of the proposed error-tracking algorithms with different amount of INTRA-refreshed MB's for sequence *Suzie* in DECT1 channel.



Fig. 13. Comparison of the proposed error-tracking algorithms with different amount of INTRA-refreshed MB's for sequence *Suzie* in DECT2 channel.

and decreases to less than 0.3 dB on average. For *Foreman*, the degradation is at most 5.2 dB, and decreases to less than 1.7 dB on average.

Fig. 11 demonstrates the robustness of the proposed error tracking algorithms with various round-trip delays for sequence *Suzie* in DECT1 channel. For smooth human speech conversations, ITU-T Recommendation G.114 specifies a maximum

round-trip delay of 400 ms. From the simulation results, we find that the proposed error-tracking algorithm still works well for a maximum delay of 500 ms. Of course, due to the inaccuracy of the linear motion model for long round-trip delay, FET2 leads to more performance degradation compared with PET. In this case, even EF MB's may be judged as being contaminated by transmission errors. However, FET2 only degrades the video quality at most 1.3 dB compared with the precise version in the case of 500-ms round-trip delay.

It is mentioned previously that INTRA-MB refreshment results in overshoot of the bit rate. However, some strategies, such as the methods described in Section II, can be applied to limit the number of INTRA-refreshed MB's for maintaining constant bit rate. Here we compare the following strategies.

- 1) PET with contamination ratio larger than K (PET (CR > K)).
- 2) PET with M MB's of largest contamination ratio, i.e., the number of INTRA-refreshed MB's with the largest contamination ratio is at most M (PET (CRN = M)).
- 3) FET with four-corner and linear motion prediction approximations and contamination ratio larger than K (FET2 (CR > K)).

The average rate-distortion relationships for sequence *Suzie* transmitted in DECT1 and DECT2 channels are shown in Figs. 12 and 13, respectively. The purpose of these experiments is to compare the performances of different methods at the same video bit rate and the same error rate. In both cases, PET (CR > 0) still has the best performance among these methods operating at the same bit rate. The PSNR results of FET2 (CR > 0) are also close to that of PET (CR > 0). Therefore, both PET and FET are robust strategies in error-prone environment.

V. CONCLUSION

To eliminate the error-propagation effects, accurate error tracking is needed in the INTRA refreshing method. We have presented a precise error-tracking coding scheme with a feedback channel for the H.263 video transmission. By utilizing the MV's generated in the regular encoding process and by tracing the backward motion dependency for each pixel, contaminated MB's are accurately tracked and the error propagation is terminated through INTRA coding the affected MB's. This approach can achieve rapid error recovery and comply with the standard. The transmitter needs neither to re-encode nor to retransmit the damaged frame. In addition, the proposed fast error tracking algorithm approximates the actual error propagation and saves considerable calculations. In short, due to its low memory requirement and computation complexity, the precise error tracking technique is especially suitable for real-time implementation.

REFERENCES

- Video Coding for Low Bitrate Communication, Draft ITU-T Recommendation H.263, May 1996.
- [2] 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*, New York, NY, Oct. 1995, pp. 475–482.

- [3] S. Lin, D. J. Costello, and M. J. Miller, "Automatic repeat error control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5–17, 1984.
- [4] S. Aign and K. Fazel, "Temporal and spatial error concealment techniques for hierarchical MPEG-2 video codec," in *Proc. ICC'95*, Seattle, WA, June 1995, pp. 1778–1783.
- [5] 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, Beijing, China, Nov. 1996, pp. 271–282.
- [6] C. C. Liao, T. H. Lee, and P. C. Chang, "Error concealment with macroblock interleaving for H.263 video transmissions," in *Proc. IPPR Conf. Computer Vision, Graphics, and Image Processing, CVGIP '97*, Tai-Chung, Taiwan, Aug. 1997, pp. 570–577.
- [7] 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, San Francisco, CA, Mar. 1998, pp. 225–232.
- [8] E. Steinbach, N. Farber, and B. Girod, "Standard compatible extension of H.263 for robust video transmission in mobile environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 872–881, Dec. 1997.
- [9] M. Ghanbari, "Postprocessing of late cells for packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 669–678, Dec. 1996.