LONG-TERM MEMORY MOTION-COMPENSATED PREDICTION FOR ROBUST VIDEO TRANSMISSION

Thomas Wiegand[1], *Niko Färber*[1,2], *Klaus Stuhlmüller*[1], *and Bernd Girod*[1,2]

[1] Telecommunications Laboratory
University of Erlangen-Nuremberg
Erlangen, Germany
[wiegand,faerber,stuhl]@LNT.de

[2] Information Systems Laboratory
Stanford University
Stanford, CA, USA
girod@ee.stanford.edu

ABSTRACT

Long-term memory prediction extends the spatial displacement vector utilized in hybrid video coding by a variable time delay permitting the use of more than one reference frame for motion compensation. This extension provides improved rate-distortion performance. However, motion compensation in combination with transmission errors leads to temporal error propagation that occurs when the reference frames at encoder and decoder differ. In this paper, we present a framework that incorporates an error estimate into rate-constrained motion estimation and mode decision. Experimental results with a Rayleigh fading channel show that long-term memory motion compensation significantly outperforms single-frame prediction.

1. INTRODUCTION

The efficiency of long-term memory motion-compensated prediction (MCP) as an approach to improve coding performance has been demonstrated in [1]. The ITU-T has decided to adopt this feature as Annex U to version 3 of the H.263 standard. In this paper, we show that the idea can also be applied to the transmission of coded video over noisy channels at improved performance.

The compressed video signal is extremely vulnerable to transmission errors. When the bit-stream received is in error, the decoder cannot or should not reconstruct the affected parts of the current frame. Rather, a concealment is invoked. But motion compensation in combination with concealment leads to temporal error propagation which causes deviating reference frames at encoder and decoder.

A popular technique to stop temporal error propagation is to encode macroblocks in INTRA mode. To invoke INTRA coding appropriately, the Error Tracking approach has been presented [2, 3], where the propagated error is tracked upon receipt of a feedback sent from decoder to encoder. Adaptive INTRA coding based on concealment distortion measures and transmission error characteristics has also been proposed for cases when no feedback is available [4, 5].

Another method to stop error propagation is to predict from reference pictures that have been confirmed to be identical at encoder and decoder. Such a scheme is included in the Reference Picture Selection (RPS) mode as described in Annex N of H.263+. RPS relies on a feedback channel [6] to acknowledge the error-free receipt of portions of the bit-stream.

In [7], multiple reference frames have been proposed for increasing the robustness of video codecs. Error propagation is modeled using a Markov chain which is used to modify the selection of the picture reference parameter employing a strategy called random lag selection. However, the actual concealment distortion, the motion vector estimation and the macroblock mode decision are not considered.

The coder control in this paper employs an estimate of the distortion between the original and the average decoding result given the statistics of the random channel and the temporal error propagation. When utilizing long-term memory MCP, temporal error propagation has to be considered in the multi-frame buffer which is controlled by the picture reference parameter. Hence, the estimate of the average decoder distortion affects the selection of the macroblock modes including the INTRA mode and motion vectors including the picture reference parameter. Experimental results with a Rayleigh fading channel show that long-term memory MCP significantly outperforms the single-frame MCP of the H.263-based anchor in the presence of error-prone channels for transmission scenarios with and without feedback.

2. THE VIDEO CODEC

The video codec employed in this paper has been published in [1]. Long-term memory MCP extends the motion vector utilized in hybrid video coding by a variable time delay permitting the use of several decoded frames for motion compensation. The frames inside the long-term memory which is simultaneously built at encoder and decoder are addressed by a combination of the codes for the spatial displacement vector and the variable time delay.

Because the video bit-stream consists of VLC words, a single bit error may cause a series of erroneous code words at the decoder. The common solution to this problem is to insert synchronization code words into the bit-stream in regular intervals. The H.263 standard supports optional GOB-headers as re-synchronization points which are also used throughout this paper. A GOB in QCIF format consists of 11 macroblocks that are arranged in one row. In case of a detected transmission error complete GOBs are discarded.

The severeness of the error caused by discarded GOBs can be reduced if error concealment techniques are employed. In our simulations, we employ simple previous frame concealment, i.e., the corrupted image content is replaced by corresponding pixels from the previous frame.

3. CODER CONTROL

The coder control employed in [1] follows the specifications of TMN-10 [8], the test model for the H.263 standard. In this paper, we also incorporate temporal error propagation.

To illustrate the effect of temporal error propagation, it is assumed that the nine GOBs of each QCIF picture are transmitted in one packet and each packet is lost with probability p or correctly received with probability $q = 1 - p$. Although the transmission scenario in the experiments does not employ such a packetization scheme, the assumption one picture to be transmitted in one packet greatly simplifies the analysis here. Figure 1 illustrates the effect of temporal error propagation in case of single-frame MCP, i.e., only the prior decoded frame can be referenced for motion compensation.

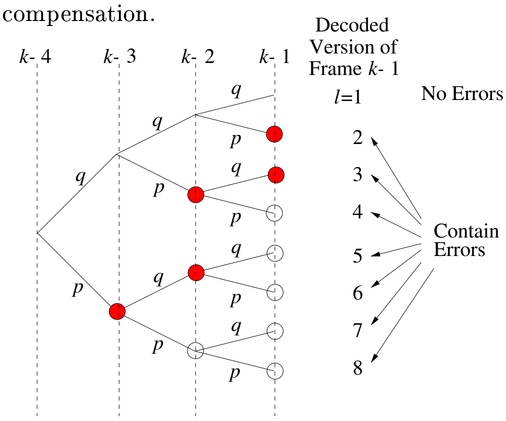

Figure 1: Binary tree of possible error events. Each node of the tree corresponds to a different decoded version of a video frame. The nodes labeled with a circle are those that contain transmission errors. The shaded circles correspond to the error cases considered in the coder control at the respective time step.

To incorporate temporal error propagation into the coder control for the frame at time instant k that references frame $k-1$, we need to estimate the average errors that have accumulated in frame $k-1$. For that, older frames than frame $k-1$ have to be considered due to temporal error propagation. For the sake of simplicity, let us assume that the frame at time instant $k-4$ is correctly decoded. In the next frame at time instant $k-3$, reference is made to frame $k-4$ using motion compensation. The frame $k-3$ is either concealed with probability p or correctly decoded with probability $q = 1-p$. Hence, the two nodes at time instant $k-3$ correspond to two different frames. The decoding of the frame $k-2$ results in 4 combinations of possible outcomes while image content in the frame $k-1$ can be decoded in 8 different ways. In general, $L = 2^k$ combinations would have to be computed for a frame that is k time instants decoded after the first frame.

If long-term memory MCP is utilized, the number of branches leaving a node in the tree of possible error events varies since frames other than just the prior decoded frame can also be referenced. Moreover, since each macroblock or block can reference a different picture in the multi-frame buffer, the tree of possible error events has to be used for each pixel. This results in $L = 2^k$ combinations per pixel after k time instants.

To obtain a computational tractable quantity for temporal error propagation, we introduce an approximation. Given the L different possible outcomes of decoding a reference picture, the average prediction error D_{AVE} reads as

$$\begin{aligned} D_{AVE}(\mathbf{v},\Delta) &= \sum_{l=1}^{L} p_l D_l \\ &= \sum_{l=1}^{L} p_l \sum_{(x,y)\in\mathcal{B}} (o[x,y] - s_\Delta^l[x+v_x, y+v_y])^2, \end{aligned} \tag{1}$$

with $\mathcal{B}$ being a 16×16 or 8×8 block, o being the original video signal, and s_Δ^l being the lth version of the reconstructed picture that is referenced via the picture reference parameter Δ. The motion vector $\mathbf{v} = (v_x, v_y)$ specifies a half-pixel accurate displacement.

Let us express the reference frame in the lth decoding branch using the correctly decoded reference frame s_Δ and the divergence between encoder and decoder ε_Δ^l. With $s_\Delta^l[x,y] = s_\Delta[x,y] + \varepsilon_\Delta^l[x,y]$, we obtain the following approximation

$$\begin{aligned} D_l &= \sum_{(x,y)\in\mathcal{B}} (o[x,y] - s_\Delta[x+v_x,y+v_y] - \varepsilon_\Delta^l[x+v_x,y+v_y])^2 \\ &\approx \sum_{(x,y)\in\mathcal{B}} (o[x,y] - s_\Delta[x+v_x,y+v_y])^2 + (\varepsilon_\Delta^l[x+v_x,y+v_y])^2 \end{aligned}$$

where we neglect the cross terms $\sum o[x,y]\cdot\varepsilon_\Delta^l[x,y]$ and $\sum s_\Delta[x,y]\cdot\varepsilon_\Delta^l[x,y]$ since we assume $o[x,y]$ and $s_\Delta[x,y]$ to be uncorrelated from $\varepsilon_\Delta^l[x,y]$ and $\varepsilon_\Delta^l[x,y]$ to have zero mean. The overall distortion term is modified to

$$\begin{aligned} D_{AVE} \approx & \sum_{(x,y)\in\mathcal{B}} (o[x,y] - s_\Delta[x+v_x,y+v_y])^2 \\ & + \sum_{l=2}^{L} p_l \sum_{(x,y)\in\mathcal{B}} (\varepsilon_\Delta^l[x+v_x,y+v_y])^2 \end{aligned} \tag{2}$$

Note that the first term corresponds to the distortion term usually computed in motion estimation routines (D_{DFD}). The second term represents the error energy caused by transmission errors. Since the computational burden evaluating (2) is still very high because of the large amount of combinations involved, we restrict the number of possibilities of errors to the two cases

1 referenced image content is in error and concealed, (branch $l = 2$ in Fig. 1)

2 referenced image part has been correctly decoded but references concealed image content (branch $l = 3$ in Fig. 1).

In Fig. 1, each node of the tree corresponds to a decoded version of a video frame. The nodes labeled with a circle are those that contain transmission errors. Our approximation incorporates only those cases with shaded circles. This approximation is motivated by assuming p to be very small with two or more successive error events in a row being unlikely. For other decoded versions s_Δ^l, we assume that the error is filtered and somewhat reduced if an error has occurred several frames in the past and is then several times motion-compensated. Nevertheless, our simplifications may be to drastic to give a reliable measure of the average decoding result. On the other hand, the intent of this work is to show how long-term memory MCP can be employed in error-prone environments and hence, the focus of this work is not on the error modeling aspect itself.

Error modeling is incorporated into motion estimation and mode decision by modifying the Lagrangian costs as

$$D_{DFD}(\mathbf{v},\Delta) + \mu D_{ERR}(\mathbf{v},\Delta) + \lambda_{MOTION} R_{MOTION}(\mathbf{v},\Delta), \tag{3}$$

with $D_{ERR}(\mathbf{v},\Delta)$ being

$$D_{ERR}(\mathbf{v},\Delta) = \sum_{l=2}^{3} \sum_{(x,y)\in\mathcal{B}} (\varepsilon_{\Delta}^{l}[x+v_x, y+v_y])^2 \tag{4}$$

and μ being a weighting term. For the long-term memory codec, the frame selection is also affected since the error modeling is incorporated into motion estimation.

Employing similar arguments as for the motion estimation, the Lagrangian costs for the INTER modes including the header bits h and the DCT coefficients c are given as

$$D_{REC}(h,\mathbf{v},\Delta,c) + \mu D_{ERR}(\mathbf{v},\Delta) + \lambda_{MODE} R_{REC}(h,\mathbf{v},\Delta,c).$$

while the Lagrangian cost for INTRA mode do not contain the error term since INTRA coding does not propagate errors by referencing a corrupted frame. The term D_{REC} refers to the distortion after decoding without temporal error propagation, the term D_{ERR} is given in (4), and R_{REC} refers to the bit-rate for the considered macroblock. Hence, one impact of the error modeling when incorporated into an H.263 and long-term memory codec is that the number of macroblocks coded in INTRA should be increased.

4. EXPERIMENTS

Our simulations employ bit error sequences that are used within ITU-T/SG16/Q.15 for the evaluation of error resilience techniques in H.263 and H.26L. The sequences are generated assuming Rayleigh fading with different amounts of channel interference, characterized by ratios of bit-energy to noise-spectral-energy (E_b/N_0) in the range of 14 to 30 dB. For channel coding we use a forward error correction (FEC) scheme that is based on Reed-Solomon (RS) codes. The bit allocation between source and channel coding can be described by the *code rate* r, which is defined as $r = K/N$. A block size of $N = 88$ is used, while the numbering of the information symbols $K \leq N$ is a free parameter to achieve code rates in the range between 32/88 and 1. In our simulations, the resulting Residual Word Error Rate (RWER) ranges from 0.3 to $6 \cdot 10^{-4}$. The RWER for a given E_b/N_0 can be reduced by approximately one order of magnitude by varying the code rate in the given range.

In our experiments, we compare **TMN-10**: the test model of the H.263 standard with Annexes D, F, I, J, T enabled and the **LTMP** coder, i.e., the long-term memory motion-compensated prediction coder with 10 frames also utilizing Annexes D, F, I, J, T. Both coders are run with a rate-control enforcing a fixed number of bits per frame when coding 210 frames of video sampled with 8.33 frames/s. For error-free transmission, the average bit-rate savings obtained by the long-term memory codec against TMN-10 are 18 % for the sequence *Foreman* when measuring at equal PSNR of 34 dB. These correspond to a PSNR gain of 0.9 dB. Similar improvements in rate-distortion performance have been measured for other sequences as well [1].

4.1. Results without Feedback

In Fig. 2, we compare the best performance in terms of maximum decoder PSNR achievable when varying the code-rate r as well as the weighting factor μ. The code-rate r is

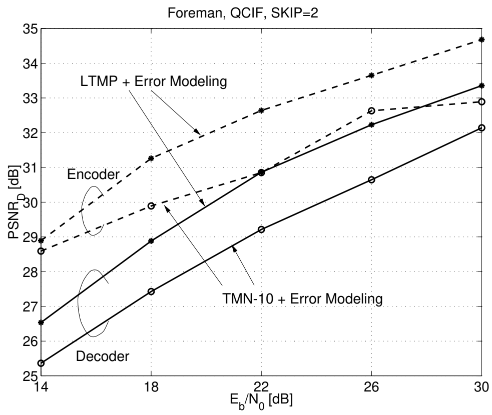

Figure 2: Average decoder PSNR vs. E_b/N_0 for the sequence *Foreman* for the optimal code-rate and error model parameter μ without feedback.

set to 8 values that are equidistantly spaced in the range 32/88 ... 1. The error modeling weight μ is varied over values $\{0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.10, 0.20, 0.30, 0.40, 0.50\}$. For each of these 88 pairs of r and μ, a bit-stream is encoded using the TMN-10 as well as the LTMP coder. The value of μ can be used to trade-off coding efficiency (small μ) and error resilience (large μ). For each bit stream, a transmission is repeated 30 times for each channel using shifted versions of the bit error sequences that correspond to $E_b/N_0 = \{14, 18, 22, 26, 30\}$. The dashed lines show encoder PSNR that corresponds to the maximum average decoder PSNR depicted with solid lines. Evaluating decoder PSNR, the LTMP coder outperforms the TMN-10 coder by 1.8 dB at $E_b/N_0 = 22$ dB.

4.2. Results with Feedback

In the following experiments, a feedback channel is utilized. The decoder sends a negative acknowledgment (NACK) for an erroneously received macroblock and a positive acknowledgment (ACK) for a correctly received macroblock. We assume that the feedback channel is error-free. The round trip delay is assumed to be approximately 250 ms, such that feedback is received 3 frames after their encoding.

In Fig. 3, we compare three feedback handling strategies for the sequence *Foreman* for the LTMP codec. The simulation conditions are similar to the previous results and the feedback handling strategies are

- **ACK mode**: long-term memory MCP is conducted as usual ($\mu = 0$) by referencing the $M = 10$ most recent decoded images for which feedback is available.

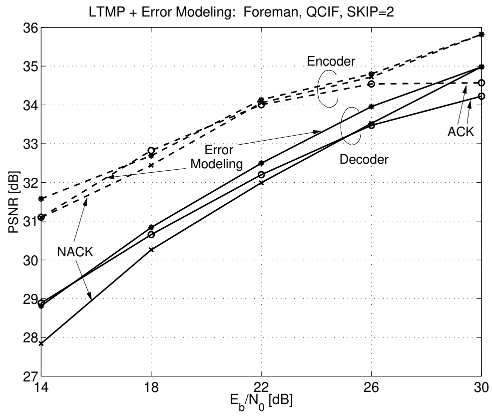

Figure 3: Average PSNR vs. E_b/N_0 for the sequence *Foreman* for the LTMP coder when feedback is utilized.

- **NACK mode**: long-term memory MCP is conducted as usual ($\mu = 0$) by referencing the most recent $M = 10$ decoded frames regardless whether or not feedback is available for them. When an error is indicated via feedback from the decoder, the depending frames are decoded again after error concealment in the feedback frame.
- **Error Modeling**: As the NACK mode, but motion estimation and mode decision are modified by the error modeling term via setting $\mu > 0$. In addition to concealing the feedback frame and re-decoding of the depending frames, also the transmission error estimate D_{ERR} is updated for those frames. For that, D_{ERR} is set to zero for the feedback frame because its decoded version at the decoder is known at the encoder. Then, the error event tree starts at the feedback frame and the transmission error estimate D_{ERR} can be updated.

The error modeling approach is superior or achieves similar performance comparing it to the ACK or NACK mode. This is because the ACK or NACK mode in the LTMP codec are special cases of the error modeling approach. The ACK mode is incorporated via large values of μ. Then, reference frames for which no feedback is available are completely avoided since for reference frames with feedback, the term D_{ERR} is set to 0. On the other hand, the NACK mode is incorporated by simply setting $\mu = 0$. Hence, the largest gain is achieved at $E_b/N_0 = 26$ dB when error modeling provides a trade-off between ACK and NACK mode.

For the TMN-10 codec, the three feedback schemes are realized in a similar way as for the LTMP coder but with the restriction to having only one reference frame for coding. Our experiments showed that the performance of the various approaches is very similar for our simulation conditions. The ACK mode shows a slight advantage for small E_b/N_0 while the NACK works better for less noisy channels.

Finally, in Fig. 4, the gains achievable with the LTMP codec over the TMN-10 codec are depicted for the feedback case. The best performance achievable for the three feedback handling strategies is depicted in both cases. We also show results for the case without feedback to illustrate the error mitigation by feedback. In the feedback case, the LTMP coder provides a PSNR gain of 1.2 dB compared to the TMN-10 coder.

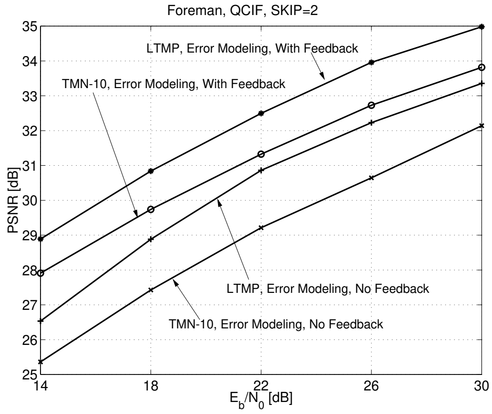

Figure 4: Average PSNR vs. E_b/N_0 for *Foreman* for the optimal feedback strategy and without feedback.

5. CONCLUDING REMARKS

In this paper we propose long-term memory prediction for efficient transmission of coded video over noisy channels. For that, the coder control takes into account the rate-distortion trade-off achievable for the video sequence given the decoder as well as the transmission errors introduced by the channel. In experiments incorporating Rayleigh fading channels, the PSNR gain at the decoder obtained for the sequence *Foreman* is 1.8 dB at $E_b/N_0 = 22$ dB for the case without feedback. When a feedback channel is available, the PSNR gain by the long-term memory scheme compared to single-frame prediction is up to 1.2 dB.

6. REFERENCES

[1] T. Wiegand, X. Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction", *IEEE Trans. CSVT*, vol. 9, no. 1, pp. 70–84, Feb. 1999.

[2] E. Steinbach, N. Färber, and B. Girod, "Standard Compatible Extension of H.263 for Robust Video Transmission in Mobile Environments", *IEEE Trans. CSVT*, vol. 7, no. 6, pp. 872–881, Dec. 1997.

[3] B. Girod and N. Färber, "Feedback-Based Error Control for Mobile Video Transmission", *Proc. IEEE*, vol. 97, no. 10, pp. 1707–1723, Oct. 1999.

[4] R. O. Hinds, T.N. Pappas, and J. S. Lim, "Joint Block-Based Video Source/Channel Coding for Packet-Switched Networks", in *Proc. VCIP*, San Jose, USA, Jan. 1998, pp. 124–133.

[5] G. Cote, S. Shirani, and F. Kossentini, "Robust H.263 Video Communication over Mobile Channels", in *Proc. ICIP*, Kobe, Japan, Oct. 1999, vol. 2, pp. 535–539.

[6] ITU-T Recommendation H.263 Version 2 (H.263+), "Video Coding for Low Bitrate Communication", Jan. 1998.

[7] M. Budagavi and J. D. Gibson, "Error Propagation in Motion Compensated Video over Wireless Channels", in *Proc. ICIP*, Santa Barbara, USA, Oct. 1997, vol. 2, pp. 89–92.

[8] ITU-T/SG16/Q15-D-65, "Video Codec Test Model, Near Term, Version 10 (TMN-10), Draft 1", Tampere, Apr. 1998.