PROGRESSIVE TEXTURE VIDEO CODING

Christian Buchner, Thomas Stockhammer[*]

Institute for Communications
Engineering (LNT)
Munich University of Technology (TUM)
80290 Munich, Germany

Detlev Marpe, Gabi Blättermann, Guido Heising[†]

Image Processing Department
Heinrich-Hertz-Institute (HHI)
for Communication Technology
10587 Berlin, Germany

ABSTRACT

In this paper, we present a novel video coding approach which provides the functionality of fine granular bitstream scalability. The proposed progressive texture video coding scheme (PTVC) consists of two parts, the current ITU-T H.26L test model for coding macroblock mode and motion information only and a new technique for progressively coding all intra and inter texture information thus providing an embedded bitstream. It can still be decoded with moderate quality degradation in case of partial loss of the bitstream. It is therefore well suited for multicast or broadcast scenarios where time varying or different fixed bitrates shall be provided to the clients as no transcoding or multiple coding process is required.

1. INTRODUCTION

In this work we present an efficient video compression scheme which supports a fast bit rate adaptation independent of the encoder. A multicast scenario in which receivers with various time-varying bit rate constraints are serviced might present a general application. Additionally any other transport network which supports prioritized data can benefit from the scalable video compression scheme which will be presented in the following. The network itself is capable to adapt the transmission bit rate without the complex and undesired process of transcoding by removing the less important part of the video bit stream.

The proposed scheme generates an embedded bitstream for each frame or, by appropriate interleaving, for each group of picture. This rate-scalability is supported by an embedded bitstream which allows decoding at multiple rates, or to be more specific at virtually any rate.

Embedded still image coding was pioneered by Shapiro in his EZW (Embedded Zerotree Wavelet) coding algorithm [1] which was later refined by Said and Pearlman in their SPIHT (Set Partitioning in Hierarchical Trees) coder [2]. Rate-scalable, embedded video coding was first proposed by Taubman and Zakhor in their LZC (Layered Zero Coding) approach [3] using 3-D subband coding with camera pan compensation. Based on this ground-breaking work, a number of embedded 3-D video coding algorithms such as in [4, 5] were proposed which combine 3-D subband coding with motion compensation for a more efficient exploitation of temporal redundancy. In order to meet more restrictive requirements with respect to delay, implementation memory and computational complexity, McCanne et al. [6] introduced a progressive video coding algorithm using simple block-based conditional replenishment for temporal and a hybrid DCT/wavelet-based transform coding approach for spatial decorrelation.

Regarding rate-scalability, conventional hybrid video coders consisting of temporal DPCM and spatial transform coding suffer from the well-known effect that the prediction state in the decoder may drift away from the encoders state in case of partial loss of bitstreams. This problem of drift prediction can be partially solved either by using multiple prediction loops or by incorporating an additional coding layer outside the temporal prediction loop. The latter approach has been adopted by the MPEG-4 video coding standard under the acronym FGS (fine granular scalability), and it consists of a base layer operating as a conventional hybrid coder and a so-called enhancement layer which progressively encodes the residue between the reconstructed base layer bitstream and the original frame by means of a pure intra coding method [7]. By ignoring any temporal relationship in the enhancement layer, coding efficiency is severely degraded when compared to the non-scalable single layer approach. However, schemes providing this kind of fine granular scalability allow to trade transmission bit rate versus quality at any point of the distribution by selecting a subset of the embedded bitstream.

Our proposed hybrid system tries to bridge the performance gap between the single layer coding approach, on the one hand, and the FGS-scheme, on the other hand, while maintaining the full functionality of an embedded approach. For this purpose, we propose a drift-compensating coding system which consists of the motion compensation apparatus of the current H.26L test model [8] and a new progressive texture coding method. We will demonstrate, that the effects of drift prediction can be significantly reduced with an appropriate intra frame update.

The remainder of this contribution is organized as follows. We will describe the proposed compression scheme in Section 2. Motion compensation, texture coding and bitstream generation will be explained. Section 3 will provide experimental results and comparisons to common video schemes. Additionally, we will discuss the performance when the transmission rate is unknown to the transmitter prior to encoding. We investigate intra coded enhancement layers as well as drift errors. Section 4 concludes the paper.

2. ARCHITECTURE

2.1. Overview

The Progressive Texture video codec (PTVC) is based on the H.26L codec [8] which has been modified to perform only motion estimation and compensation. Coding of transform coefficients has

[*] e-mail: {buchner, tom}@lnt.ei.tum.de, Tel: +49 89 28923474
[†] e-mail: {marpe,blaetter,heising}@hhi.de, Tel.: +49 30 31002619.

been disabled in the H.26L codec. Instead, we use an embedded bitplane coding to code I-frames and the residual error resulting from motion compensation. The structure of the resulting codec is shown in figure 1. The resulting video bitstream consists of the interleaving of the H.26L component and the progressive texture bitstream. Therefore, the texture bitstream can be truncated at any point. We determine a constant bitrate at which we feed back the resulting image into the hybrid predictive loop of the H.26L coder. We refer to this bitrate as feedback bitrate r_f. In the following we will briefly discuss the motion compensation in H.26L and then present the progressive texture coding in more detail. For an extensive presentation of the PTVC we refer to [9].

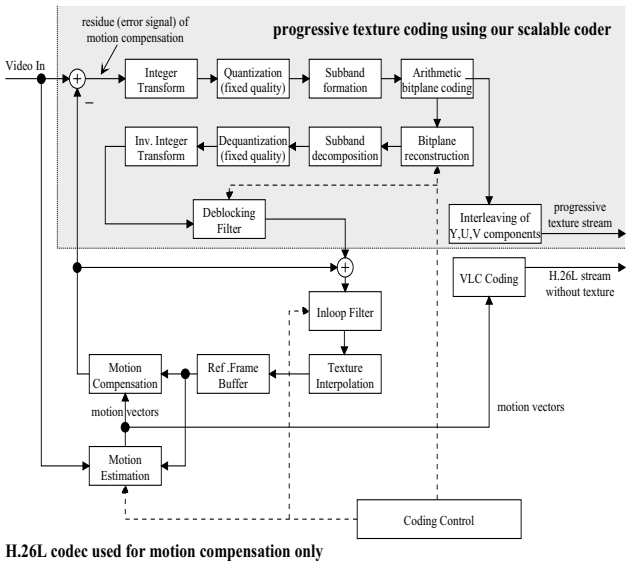

Fig. 1. Architecture of the Progressive Texture Video Codec

2.2. Motion Estimation and Compensation

The proposed system uses the motion estimation and compensation apparatus of the H.26L codec TML5.0 [8]. H.26L uses a block based motion compensation with variable vector block sizes for each macroblock. The vector blocks can be of square or rectangular shape. Motion estimation uses 1/4 Pel accuracy and performs an exhaustive search over all integer Pel positions within the search range. Furthermore, multiple reference frames (up to 5) are possibly used in motion estimation, allowing a more accurate prediction. For more details we refer to [8] and [9].

2.3. Progressive Texture Coding

The residual error of motion compensation (the displaced frame difference, DFD) is formed in the spatial domain. The DFD is what we refer to as texture information. The main conceptual features of our bitplane coder operating on the texture information can be summarized as follows:

- Distinction between significance and refinement bits,
- Context-based arithmetic coding of the binary decisions,
- Several passes for each bitplane.

The same 4×4 block based integer transform as used in the H.26L test model [8] is also employed for our embedded texture coding approach. However, before encoding the coefficients are rearranged into 16 corresponding subbands, such that, for example, all DC coefficients are contained in the upper left subband shown in Fig. 2. For each bit encountered in the bitplane representation of the absolute values of a given coefficient, a distinction is made between significance and refinement bits. Significance bits are those bits indicating whether a given coefficient has already become significant, i.e. whether its most significant bit (MSB) has already shown up in a given bitplane. Every bit below the MSB is interpreted as a so-called refinement bit (cf. Fig. 3). The information of the current state of the coefficients will be managed in a binary significance map (one for each subband), as shown in Fig. 3. This map represents the currently available significance information of the amplitudes.

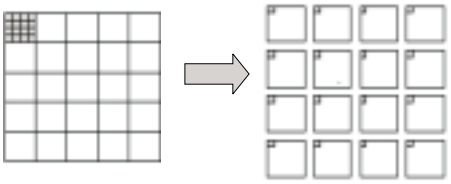

Fig. 2. The spatially local block based representation of texture is converted into a spatially global subband oriented representation.

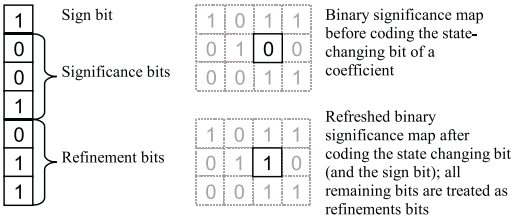

Fig. 3. The bitplane representation of a single coefficient (left) and its position in the corresponding significance map (right).

By means of the significance map it is possible to exploit the remaining spatial correlations of the significance information within a subband with the instrument of adaptive context-based arithmetic coding. In contrast to significance bits, refinement bits usually show only weak spatial correlations and could be considered as uniformly distributed. The underlying concept of partitioning a representation of transform coefficients into sub-sets with different statistical properties has been successfully employed in both scalable and non-scalable coding approaches [10, 11]. The bitplane coding process is divided into three different scans of each bitplane in order to sort the information with respect to the expected gain in a rate/distortion (R/D) sense, which obviously is not directly related to the spatial position of a given bit within a subband. This method increases the granularity of the bit-stream because more optimal truncation points are generated. In the context of JPEG-2000 [10] it is known as the concept of fractional bitplanes.

The first scan operates on non-singular significance information, that means, in this scan only those bits will be coded, which are significance bits and which have at least one significant neighbor. Here, the significance coding routine and, if a MSB is encountered, the sign coding routine are used. In the second scan the refinement bits are coded using the refinement routine. The third scan finally collects all remaining bits, which have not been

coded, by the two preceding scans using the same coding routines as in the first pass. Note, that all bits of the coefficients are visited and examined in raster scan order within a given subband. The actual coding, however is performed according to the scan conditions. The processing of the subbands is performed in global zigzag scan within each bit plane. The context formation for encoding of significance and sign bits is done by mapping a neighborhood of the related coefficient to a reduced number of context states. Fig. 4 illustrates the neighborhood for both coding primitives.

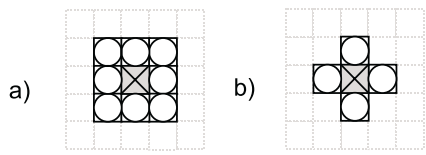

Fig. 4. a) 8-neighborhood used for encoding of significance bits, and b) 4-neighborhood for context formation of sign bits

Coding of significance bits is performed by using a context which depends on the significance state of the local 8-neighborhood. The 256 states are quantized to three final contexts which describe the activity of the neighbors (no, low, high activity). Sign bits are coded with a context depending on sign and significance states of the local 4-neighborhood. This routine is only invoked, when a coefficient indicates its significance for the first time. Refinement bits are coded with one single statistical model.

2.4. Embedding of Bitstream and Rate Control

The coding methods described previously provide many possibilities to generate an embedded bitstream with appropriate features. Especially the regular introduction of I-frames to compensate drift effects when parts of the bitstreams are lost is a crucial task. Additionally, the allocation of bits within the frames and color components can be adjusted. And finally, the generation of embedded transmission packets has to be specified. The rate allocation might be done in a rate-distortion sense including delay and buffer constraints as well as transmission rate characteristics.

However, to generate appropriate experimental results we have simplified the rate control as follows. The frame rate f_r is assumed to be constant, the I-frame period P_I is also constant and the number of bits within on group of pictures is constant according to the selected bit rate r_b. We assume that we generate exactly one transmission packet per frame with length $L_p = r_b/f_r$. Additionally, the ratio of the size of I-frames and P-frames R_{IP} is fixed according to a maximum tolerable delay. The residual frames within one GOP are interleaved such that more important information is always packetized before less important information but the maximum delay is never exceeded. For the intra-coded enhancement layer, we do not distinguish between residue of I-frames and P-frames. The color components are also transmitted with a fixed ratio R_c which is assured by appropriate interleaving. Figure 5 shows the layout of the PTVC bitstream. For more details on the bitstream generation we refer to [9].

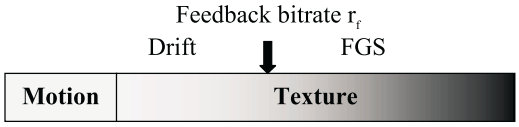

Fig. 5. Layout of the PTVC bitstream.

3. PERFORMANCE OF THE PROGRESSIVE TEXTURE VIDEO CODEC

3.1. Test Conditions

Two set of experiments have been performed to evaluate our proposed rate-scalable video coder. In our first simulation, we compared the performance of the PTVC scheme to that of two non-scalable conventional hybrid coders for the case of *a priori* known bitrate by adjusting the feedback bitrate r_f to an appropriately fixed value in our en- and decoder. To demonstrate the additional functionality of our approach, we show in the second part of our experiments how rate-scalability with and without drift can be achieved by parameterizing the feedback bitrate of our proposed scheme. All simulations were carried out using the QCIF test sequence *Foreman* (30 Hz, 300 frames, 176×144 pels) at a constant frame rate of $f_r = 10$ Hz.

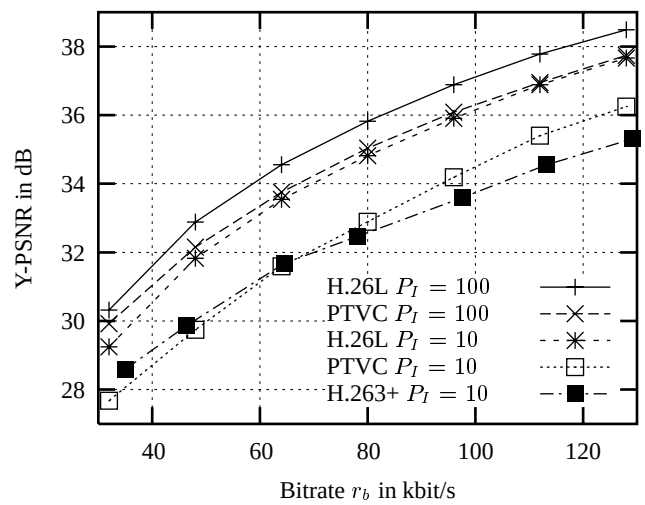

Fig. 6. Performance of PTVC for known bit rate.

3.2. Performance for Known Transmission Bitrate

As reference systems we used TML5 [8] of H.26L and TMN9 of the ITU-T Rec. H.263+ [12], the latter with advanced options of Annexes D, F, I, J and T. To achieve a given target rate, a simple off-line rate control mechanism was used for both reference schemes, whereas for PTVC the rate control described in Section 2.4 was employed with ratios $R_{IP} = 6 : 1$ and $R_c = 10 : 1 : 1$. This ratio is similar to the one of the reference codecs. Fig. 6 shows the results of our experiments using two different I-frame periods $P_I = 10, 100$ with one I-frame every 10th and every 100th frame, respectively. As can be seen from the graph, our new texture coding is only little inferior to the coding method of H.26L,[1] at least in the case where only one I-frame ($P_I = 100$) for the whole sequence is coded. Due to an efficient but inherently non-scalable spatial prediction scheme as part of its I-frame coding method (*cf.*[8]), the H.26L test model provides a more distinctive gain when more I-frames are inserted ($P_I = 10$). However, even in this scenario, PTVC has a rate-distortion performance similar to or better than H.263+ with advanced coding options. Currently, we are investigating how to improve the coding efficiency of our progressive texture coding method especially for I-frames.

[1] Note, that TML5 and PTVC share the same motion model.

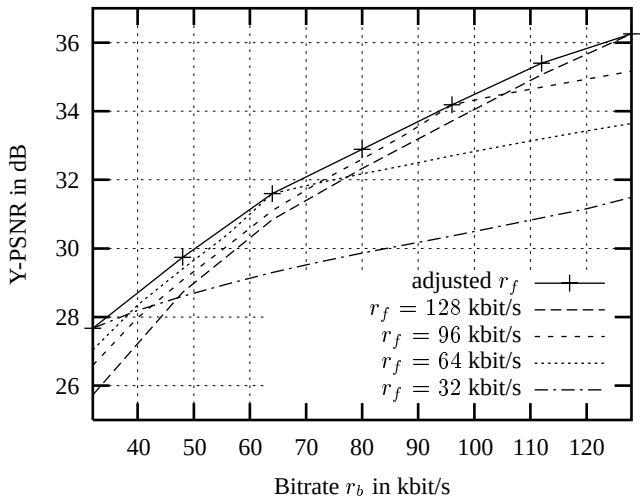

Fig. 7. Performance of PTVC for known and unknown bit rate with different feedback bitrates.

3.3. Performance for Unknown Transmission Bitrate

In this section we provide results which show the benefits of our proposed coding scheme. Assume that we have a sequence which was coded without knowing the transmission bit rate. Therefore, neither the rate in a regular video codec nor the feedback bitrate for the PTVC can be adjusted properly. However, due to the progressive texture coding a reduced transmission rate can be compensated by transmitting just the first part of the transmission packets. In the case where the transmission rate is greater than the applied feedback bitrate r_f we just cut the intra-coded enhancement layer similar to the MPEG-4 FGS approach. If the transmission rate is below the feedback bitrate, a drift effect occurs as encoder and decoder have different reference frames. However, due to the drift compensating I-frame update this error can be reduced significantly. Figure 7 shows the performance of the PTVC for different feedback bit rates. The parameters for the PTVC are equivalent to the one presented in Section 3.2 with I-frame period $P_I = 10$. The transmission bit rate is assumed to be constant such that each transmission packet is truncated at the bit position r_b/f_r. It is worth to mention that any transmission bit rate and therefore any bit position can be selected due to the progressive texture coding.

For a low feedback bitrate the performance increase for increasing transmission bit rate is rather small. This is due to the intra-coding of the enhancement layer. Similar performance is reported by the MPEG-4 FGS approach. For high feedback bitrate and slightly lower transmission bit rate it can be observed that the quality loss due to the drift is visible but not significant. Comparing the lowest feedback bitrate at $r_f = 32$ kbit/s and the highest feedback bitrate at $r_f = 128$ kbit/s only for an unknown transmission bit rate below $r_b \approx 45$ kbit/s it would beneficial to use the intra-coded enhancement layer approach. Therefore, if drift compensating mechanisms like a regular intra update are performed, the scheme with high feedback bitrate results in much better performance than the FGS approach for almost all transmission bit rates. The objective results based on the PSNR can be verified by subjective observations.

4. CONCLUSIONS

We have presented a new video coding scheme which combines the ITU-T H.26L test model with a progressive texture coder utilizing context based arithmetic encoding of bitplanes in the frequency domain. It has been shown that due to the embeddedness of our approach moderate quality degradation occurs when only parts of the bitstream are decoded. In this case drift prediction is introduced which can be considerably reduced by a periodic I-frame update. Experimental results indicate the usefulness of our approach for streaming video applications over networks with time-varying bandwidth. Our proposed scheme can also be viewed as a means for data partitioning and therefore should be combined with unequal error protection or prioritization mechanisms to achieve a higher robustness in error prone environments. In [13] an appropriate system for packet erasure channels is presented. Significant gains with up to 5 dB in PSNR compared to standard approaches for packet erasure channels are reported. In addition the new approach will benefit from an adaptation of the bitstream to the conditions of the underlying network. Therefore, future work will focus on error resilience and network adaptation aspects as well as improvements of coding efficiency, like encoder optimization techniques or improved entropy coding.

5. REFERENCES

[1] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, December 1993.

[2] A. Said and W.A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, pp. 243–250, June 1996.

[3] D. Taubman and A.Zakhor, "Multirate 3-D subband coding of video," *IEEE Transactions on Image Processing*, vol. 3, pp. 572–584, 1994.

[4] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3D SPIHT)," *IEEE Transactions on Circuits and Systems for Video Techn.*, December 2000.

[5] S.-T. Hsiang and J. W. Woods, "Embedded video coding using motion compensated 3-D subband/wavelet filter bank," in *Proceedings Packet Video Workshop*, Sardinia, Italy, May 2000.

[6] S. McCanne, M. Vetterli, and V. Jacobson, "Low-complexity video coding for receiver-driven layered multicast," *IEEE Journal on Selected Areas in Communication*, vol. 15, pp. 983–1001, August 1997.

[7] ISO/IEC 14496-2, "MPEG-4 video FGS v.4.0," Tech. Rep. N3317, Proposed Draft Amendment (PDAM), Noordwijkerhout, the Netherlands, March 2000.

[8] Gisle Bjontegaard, *H.26L Test Model Long Term Number 5 (TML-5) draft 0*, ITU-T Standardization Sector, Oct. 2000, Doc. Q15-K-59d1.

[9] Christian Buchner, "Progressive video coding for error-prone channels," M.S. thesis, Institute for Communications Engineering, Munich University of Technology, October 2000.

[10] ISO/IEC CD 15444-1, "JPEG-2000 image coding system," Tech. Rep., Committee Draft, Version 1.0, December 2000.

[11] D. Marpe and H. L. Cycon, "Very low bit-rate video coding using wavelet-based techniques," *IEEE Transactions on Circuits and Systems for Video Techn.*, vol. 9, no. 1, pp. 85–94, February 1999.

[12] ITU-T Recommendation H.263 Version 2, "Video coding for low bit-rate communication," Tech. Rep., Jan. 1998.

[13] C. Buchner and T. Stockhammer, "Progressive texture video streaming for lossy packet networks," submitted for Packet Video Workshop 2001, Kyongju, Korea.