FEEDBACK LOOP FOR CODER CONTROL IN A BLOCK-BASED HYBRID CODER WITH MESH-BASED MOTION COMPENSATION

Jörn Ostermann

AT&T Labs - Research, 101 Crawfords Corner Road, Holmdel, NJ, USA

email: ostermann@research.att.com

Abstract

In this paper, the motion estimation and motion compensation within a block-based hybrid coder is modified. Input to the motion estimator is the original frame and a representation of the previously decoded frame generated by means of a second prediction loop. The second prediction loop works in parallel to the prediction loop of the decoder. It distinguishes itself from the conventional coder prediction loop such that for blocks with transmitted DCT coefficients, the original image signal is fed into the second prediction loop, and for blocks without transmitted DCT coefficients, the motion compensated signal is fed into the prediction loop. Such, the image generated by the second prediction loop is influenced by motion but not by the quantization noise of the DCT. The motion-compensated prediction image from the second predictor loop can be easily used for control of the encoder. This coder control is not influenced by the actual quantization selected by the encoder and hence very stable for a wide range of bitrates. Savings of 15% to 25% in bitrate can be observed without loss of subjective picture quality.

1. INTRODUCTION

In order to adapt a block-based hybrid coder like H.263 to a constant bitrate, the DCT-coefficients are quantized. Whereas a coarse quantization results in poor picture quality and low bitrate, a fine quantization results in good image quality and high bitrate. Alternatively, encoding of the prediction error in some blocks can be suppressed. Frequently, the quantizer step size is mainly influenced by the bitrate available for coding the remainder of the currently encoded frame. The bitrate spent for one macro block then depends on the prediction error of the macro block and on the quantizer step size.

Advanced coder control mechanisms are trying to quantize the DCT coefficients according to the image contents by doing an image analysis which decides where the important parts of the image are. Some researchers propose to code those blocks of an image with the highest prediction error [6]. For video conferencing, researchers try to locate roughly the position of the face of the speaker and assign a smaller quantizer to the face area [7]. However, this approach relies on the recognition algorithm. In situations where the recognition fails, this kind of coder control actually decreases subjective picture quality. All these schemes are not able to decide whether the prediction error signal is due to coding errors propagated from previously coded images or due to changes in the image content. Especially when coding image sequences at low data rates, the propagated coding errors can be as eminent as the prediction errors due to changes in the image content. Thus, a coder control evaluating changes in image content usually evaluates the original image sequence and defines the areas for which the prediction error has to be updated. This methods bears the inherent danger, that there is a drift between the coder control and the coder itself.

In this contribution, a method is proposed to allow a separation of prediction errors occurring due to changes in the scene and prediction errors due to previously induced coding errors with a coarse quantization step size. This is achieved by means of a second prediction loop in the encoder as described in Section 2. Coder control is discussed in Section 3. Experimental results are presented in Section 4.

2. FEEDBACK CONTROL USING A SECOND PREDICTOR LOOP

Figure 1 shows the block diagram of a block-based hybrid coder with motion compensation and a second predictor loop. As the reader can easily see, the top part of that figure shows a conventional hybrid coder with motion compensation that is just augmented by a switch s_I. This switch is controlled by coder control. If coder control closes the switch, the prediction error signal e_I of the current block is encoded and transmitted, otherwise, the prediction error is not coded. It is assumed that the predictor contains the frame memory with the previously coded image.

0-8186-7919-0/97 $10.00

At the bottom part of Figure 1, a second prediction loop without DCT, Quantizer and IDCT is shown. Predictor 1 and predictor 2 are identical. Whenever the prediction error signal e_1 is coded, the prediction error signal e_2 is added to the motion compensated prediction thus providing the original image signal for that particular block at the input of predictor 2. If e_1 is not coded, the motion compensated signal is sent to the input of predictor 2. Hence, the frame memory of predictor 2 contains the original image signal for blocks which have been encoded and transmitted to the receiver and it contains the motion compensated prediction for the not-coded blocks.

Coder control looks at the prediction error signals e_1 and e_2. Whereas e_1 is due to limitations of motion compensation and quantization errors of previously quantized blocks, e_2 is only due to temporal changes in the video sequences and limitations in motion compensation. Now, coder control is able to distinctly decide which blocks of the current image require an update of the prediction error signal due to

- current changes in the image sequence,
- loss of signal quality because of repeated prediction using fractional pel motion compensation,
- large quantization errors introduced when coding previous images.

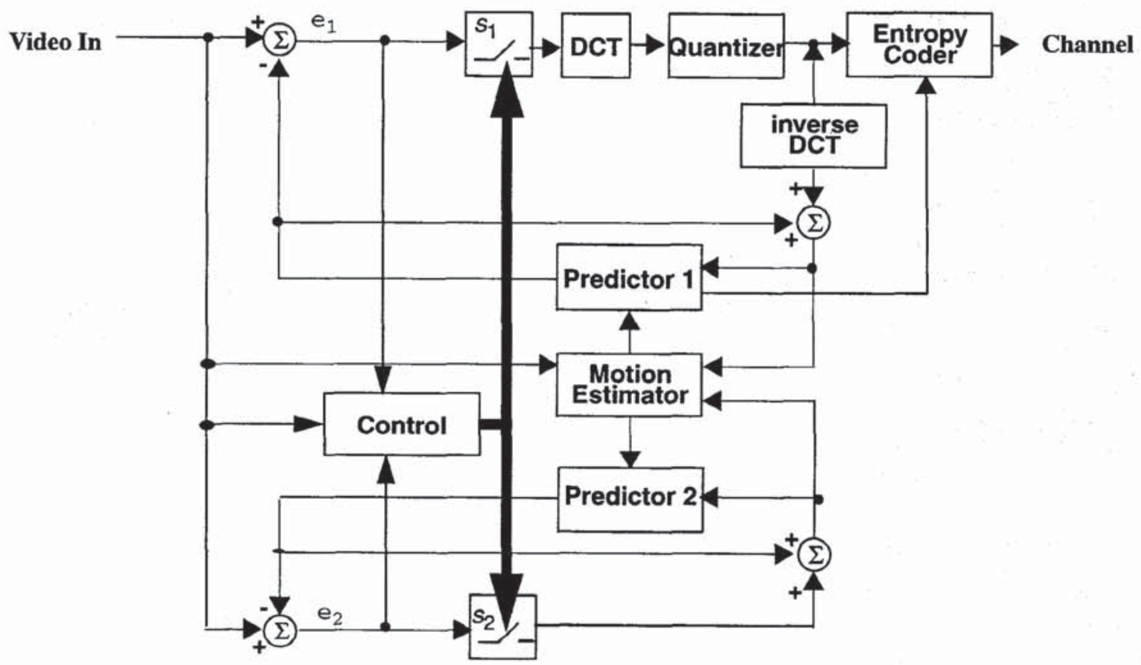

Figure 1: Block diagram of a block-based hybrid coder with a second predictor loop. The switches s_1 and s_2 are operated synchronously by coder control. Predictor 1 and Predictor 2 use the same prediction mode for each block.

3. CODER CONTROL

A TMN4-based coder [1] with a constant quantizer stepsize and a mesh-based motion compensation [5][8] has been implemented. The coder control performs a simple change detection on the input images in order to detect moving objects [3]. In order to find changes in the image that cannot be described by the displacement vector field, an algorithm similar to the detection of model failures as known from object-based analysis-synthesis coding is applied to the prediction error signal e_2 [2]: A noise adaptive threshold is applied to the prediction error signal e_2. In order to retain only clusters of points with an error signal above the noise adaptive threshold, the binary image is median filtered and smoothed using a morphological filter. Remaining blobs of points are considered to be important if their size exceeds a threshold like 50 pels for CIF images. The blocks affected by these blobs are marked for texture update.

Coder control operates the switches as follows:

1. $s_{1,2}$ are closed for an 8x8 block if it contains changed and unchanged areas according to the change detection mask.
2. $s_{1,2}$ are closed for an 8x8 block if it is marked for texture update according to the noise adaptive thresholding and filtering.
3. otherwise, $s_{1,2}$ are open.

Please note, without the second predictor loop, only prediction error signal e_1 is available. In this noisy error signal, areas not well predicted by motion compensation due to an estimation error or due to the failure of the underlying source model are difficult to segment in a meaningful way from badly predicted areas due to quantization nose.

4. EXPERIMENTAL RESULTS

For this experiment, TMN4 (1 motion vector for each 16x16 block) was modified in two steps: Rate control: The coder operated with a fixed quantizer stepsize of 11. The frame rate is constant, 10Hz. Motion estimation: Half pel accuracy, full search as specified in TMN4. The coder with the above modifications is called TMN4ref. Based on TMN4ref, three extensions were developed.

1. TMN4_seg: TMN4ref with change detection [3] as coder control preventing update of blocks in static areas.
2. TMN4_seg_motinthier: TMN4_seg with hierarchical motion estimation [4], mesh-based interpolation of the motion vectors, coder control according to Section 2 analyzing prediction error signal e_1.
3. TMN4_seg_motinthier_2P: TMN4_seg with hierarchical motion estimation [4], mesh-based interpolation of the motion vectors, second prediction loop and coder control according to Section 2 analyzing prediction error signal e_2.

For the MPEG-4 test sequence *Akiyo* and *Mother&Daughter*, the bitrates required for coding are given in Table 1 and Table 2 . Informal subjective evaluation have shown that the picture quality is the same for the coder TMN4ref and TMN4_seg_motinthier_2P. As can be seen, not updating the static background (TMN4_seg) gives a no change for *Akiyo* (noise free static background) and a slight decrease in bit rate and PSNR for *Mother&Daughter*. The segmentation of the prediction error signal e_2 is much more effective in conserving bits than the segmentation of e_1. The segmentation of the prediction error e_2 allows to achieve the same picture quality as TMN4ref and TMN4_seg_motinthier - but at a 15% to 25% lower bitrate. Please note that adaptive block size for motion compensation and overlapping motion compensation as used in H.263/TMN5 can also be integrated into this coder.

Table 1: Bitrates and PSNR for encoding the test sequence *Akiyo* at CIF, 10 Hz with a constant quantizer stepsize of 11.

Coder	Bitrate	PSNR
TMN4ref	47.88	37.12
TMN4_seg	47.72	36.93
TMN4_seg_motinthier	50.66	36.54
TMN4_seg_motinthier_2P	41.09	35.65

Table 2: Bitrates and PSNR for encoding the test sequence *Mother&Daughter* at CSIF, 10 Hz with a constant quantizer stepsize of 11.

Coder	Bitrate	PSNR
TMN4ref	69.42	37.96
TMN4_seg	63.42	37.62
TMN4_seg_motinthier	60.03	37.05
TMN4_seg_motinthier_2P	52.00	36.33

5. CONCLUSIONS

A mesh-based coder is presented which actually outperforms a purely block-based coder. This became only possible due to a coder control requiring a second predictor loop at the encoder. The second predictor loop is only influenced by motion in the sequence and the update of the prediction error. Hence, the coder control is not disturbed by quantization noise but only by the effects of motion compensation using the applied motion model. For scenes with a static background, this coder control allows savings in bitrate of 15% to 25% without subjective loss in picture quality.

6. REFERENCES

1. ITU-T Study group 15, Working Party 15/1, TMN-4, Doc. LBC-94-124.
2. J. Ostermann: "Object-based analysis-synthesis coding based on the source model of moving rigid 3D objects", Signal Processing: Image Communication, No. 6, pp. 143-161. 1994.

3. Hoetter, M., Thoma, R.: "Image segmentation based on object oriented mapping parameter estimation", Signal Processing, vol. 15, no. 3, pp. 315-334, October 1988.
4. T. Ebrahimi, H. Chen, B. Haskell, "Joint motion estimation and segmentation for very low bitrate video coding", SPIE Visual communications and image processing, 1995.
5. Y. Altunbasak, A.M. Tekalp, "Two-dimensional object-based coding using a content-based mesh and affine motion parameterization, ICIP-95, TP3.02, Washington DC, October 1995.
6. K. Grotz, J. Maier, G. Suessmeier, "A 64 kbit/s videophone codec with forward analysis and control", Signal Processing: Image Communication, Vol. 1, No. 2, pp. 103-116, October 1989.
7. E. Badique, "Knowledge-based facial area recognition and improved coding in a CCITT-compatible low-bitrate video-codec", Proc. Picture Coding Symposium 1990, Boston, 1990.
8. Y. Wang, J. Ostermann, "Evaluation of mesh-based motion estimation in H.263 like coders", submitted for publication to Video Communications.

Figure 2: Frame 90 of the sequence *Mother&Daughter* (top) and Frame 76 of *Akiyo* (bottom) coded with TMN4 (left) and with the mesh-based coder using the second predictor loop for coder control (right). The image quality does not change although the proposed coder (right) uses between 15% and 25% less bits.