Transcoding of MPEG-2 Video Bitstreams in the Frequency Domain

Bernd Sostawa and Joachim Speidel
Institut für Nachrichtenübertragung, University of Stuttgart

ABSTRACT — MPEG-2 video coding is widely used in broadcasting and increasingly in studio applications. For reasons of limited storage and/or transmission capacity, it may be necessary to reduce the bit rate of MPEG-2 video bitstreams. We present a frequency domain transcoder (FDTC) with low complexity and low memory requirements which can reduce the bit rate of the incoming video bitstream. A rate control with low delay provides a constant bit rate (CBR) stream at the output, independent whether the input bit rate is variable (VBR) or constant (CBR). The FDTC achieves the same peak signal-to-noise ratio (PSNR) as a bulky cascade of a complete MPEG-2 video decoder and encoder. Moreover, for real time applications, an implementation of the FDTC on a media processor is presented.

1 Introduction

MPEG encoded video signals are widely used, e.g. on storage media for digital TV and Internet applications, as well as for digital video and multimedia broadcasting (DVB, DMB). MPEG-2 video bitstreams with a high bit rate and high quality are stored on video servers that can be accessed by the users. If the bit rate of the stored video bitstreams exceeds the capacity of the transmission channel between server and user, or if the storage capacity at the user's side is limited, a transcoder is required to reduce the bit rate. In a video on demand (VoD) or pay per view application, the users can be provided with different bit rates depending on the picture quality they are willing to pay for. Also in a car, where several digital video signals have to be distributed to the rear passengers over a communication bus with limited capacity, a transcoder is required [1], [2].

Fig. 1 shows the various system configurations in which a transcoder can be used. In this paper we use the term transcoder to identify a system that reduces the bit rate of an input video bitstream with rate r_1 to an output bitstream with rate $r_2 < r_1$. Both digital signals shall comply with the MPEG-2 standard [3]. The transcoder receives the bitstream with a high bit rate from a storage system or a wideband transmission channel. The storage system may be a hard disk drive like in the video server example above, a tape drive or a DVD drive. The transcoded bitstream is stored on smaller memory or is transmitted via a narrowband transmission channel, both at lower costs.

However, the various storage media and transmission channels use bitstreams with different bit rate characteristics. A tape drive, e.g. a potential digital video tape recorder, uses CBR bitstreams whereas we find VBR bitstreams on a DVD. Many transmission channels use CBR streams, provided that no statistical multiplexing is used. The Internet and ATM networks prefer VBR signals. The design of the transcoder depends on the bit rate characteristic at the input and output. The most relevant transitions are CBR to CBR and VBR to CBR, i.e. in both cases the transcoder's rate control has to provide a CBR bitstream with a lower bit rate at the output.

This paper is organized as follows: Section 2 describes the FDTC and its processing of the DCT coefficients. In Section 3, the rate control algorithms for CBR to CBR as well as VBR to CBR transcoding are introduced. Some results are provided in Section 4. Section 5 shortly presents the real time implementation of the transcoder on a media processor. Finally, this paper is concluded in Section 6.

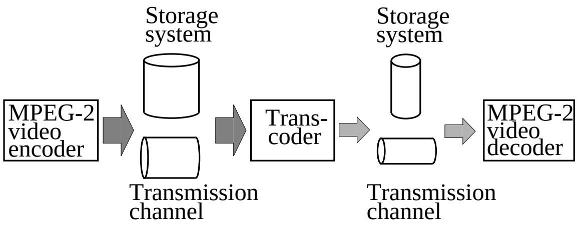

Fig. 1. System configurations in which a transcoder can be used. The width of the arrows indicates the size of the bit rate.

8. Dortmunder Fernsehseminar; Sept. 1999; Multimedia: Anwendungen, Technologie, Systeme; ITG-Fachbericht 156; VDE Verlag 1999; pages 197-202

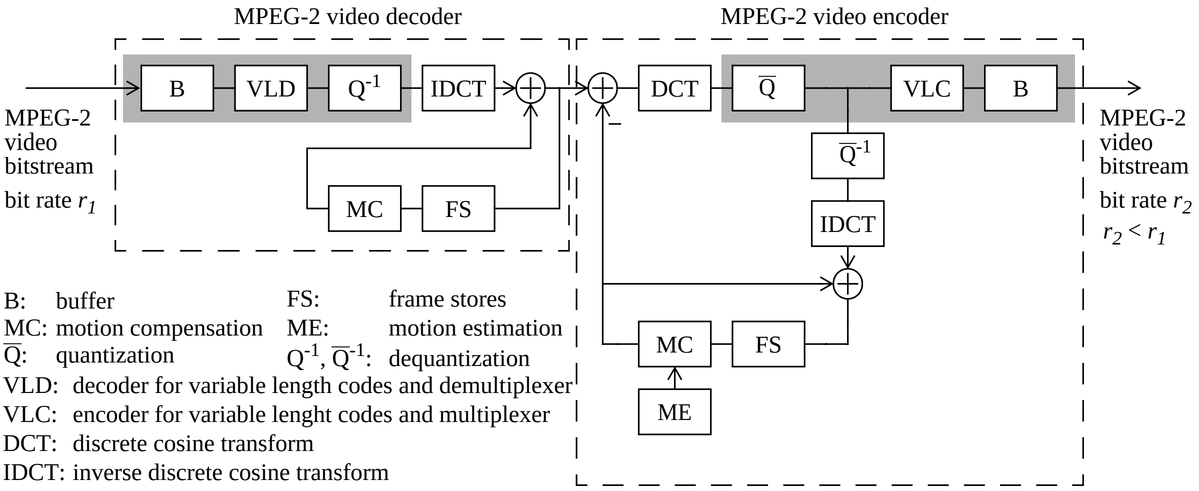

Fig. 2. The General Transcoder.

2 The Frequency Domain Transcoder (FDTC)

2.1 The Structure of the FDTC

In order to develop the concept of the Frequency Domain Transcoder (FDTC), **Fig. 2** shows a straightforward solution of a transcoder which is a cascade connection of a complete MPEG-2 video decoder and an encoder. Subsequently, we call this arrangement the General Transcoder. Obviously, this solution is quite complex, and thus has a high implementation cost. One might think, that due to the full-fledged decoding and encoding stages in Fig. 2 the transcoding process would result in an optimum picture quality compared to simpler solutions. However, we show that a more efficient scheme exists which is even less bulky than the General Transcoder and which provides similar picture quality. To develop such a structure, we first outline some observations concerning Fig. 2. Obviously, there is no communication between the decoder and the encoder. As a consequence, the encoder does not use the original coding parameters of the input bitstream with bitrate r_1 which are available in the decoder. Hence, the encoder has to remake all coding steps and decisions including the motion estimation and compensation. This is done on the basis of the decoded picture which exhibits coding noise.

The efficiency of the General Transcoder can be increased significantly by introducing a communication link between decoder and encoder. By that means, the encoder can partly adopt the coding parameters of the input bitstream with bitrate r_1. This method prevents from a calculation of all coding parameters by the encoder.

Investigations have been done concerning requantization of DCT coefficients and motion vectors [1]. As a result, manipulation of motion vectors for bit rate reduction is less efficient than requantization of DCT coefficients. For that reason, the entire motion information of the input bitstream with bitrate r_1 is fed directly into the outgoing bitstream with bit rate $r_2 < r_1$ and no motion estimation is carried out. As a consequence, the complexity of the transcoder is further reduced. The FDTC only consists of the gey-shaded blocks in Fig. 2 and is shown in more detail in **Fig. 3**. A rate control adjusts the new quantization factors to match the desired output bit rate. As the requantization is done with DCT coefficients, the signal processing entirely takes place in the frequency domain.

The FDTC has several advantages. No motion estimation is required because the motion vectors of the input bitstream are reused, thus reducing the computational cost. Since the motion compensation in a feedback loop is omitted, no frame stores are needed any longer. Hence, the memory requirements are low. However, the missing feedback loop has a consequence. The errors

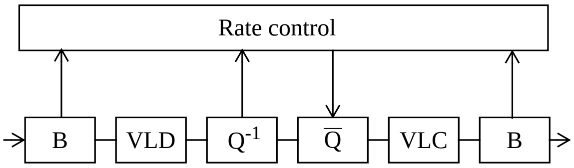

Fig. 3. The Frequency Domain Transcoder (FDTC). For the abbreviations see Fig. 2.

introduced by the requantization are not incorporated in the transcoded bitstream. This leads to a slight drift of the non-intra pictures at the transmitter and receiver. However, we found that the drift is hardly perceptible due to the balancing influence of a properly designed rate control.

2.2 The Processing of the DCT Coefficients

As shown in Fig. 3, the DCT coefficients are dequantized within the block Q^{-1} with the quantiser_scale received in the input bitstream. Then, requantization with a higher quantiser_scale takes place in the block $\overline{Q}$. With the coarser requantization, more coefficients are set to zero and the absolute value of the remaining non-zero coefficients becomes smaller. The result is a lower bit rate $r_2 < r_1$.

With intra blocks, the first DCT coefficient representing the DC value remains unchanged. Only the subsequent DCT coefficients are requantized. With non-intra blocks, both the first and the subsequent DCT coefficients are requantized. If the requantization factor is sufficiently high, all DCT coefficients vanish. Then, the coded block pattern has to be adjusted. In some cases skipped macroblocks can be introduced.

Another strategy for the processing of the DCT coefficients is to set a predetermined number of high frequency coefficients to zero [1]. Although the implementation is simple, only the requantization method is considered in this paper.

2.3 Skipped B-Pictures

As B-pictures are not used for prediction, it is possible to skip macroblocks within such pictures without affecting the picture quality of subsequent pictures. If all admissible macroblocks of a B-picture are skipped and the remaining non-skipped macroblocks are coded with a zero forward motion vector but without macroblock_pattern, the picture content of the preceding I- or P-picture, respectively, will be repeated giving the impression of a skipped B-picture. With this procedure, the bit rate is strongly reduced because both the coefficient and motion data are removed from the bitstream. Especially the removal of the motion vector data reduces the number of bits because B-pictures carry a comparatively large amount of motion information.

Skipped B-pictures reduce the effective frame rate which is often perceived as a degredation of quality. Therefore, this method should only be used if the target bit rate is extremely low or if there is an impending underflow of the video buffering verifier (VBV) buffer.

3 Rate Control

In order to match the desired bit rate for the transcoded bitstream, a rate control has to adjust the requantization factors. The FDTC produces a CBR bitstream at its output. Two rate control methods are proposed in the following. The first method handles only CBR input bitstreams whereas the second one can operate with CBR as well as VBR input bitstreams.

3.1 CBR to CBR Transcoding

The proposed rate control is based on a principal method that can be used to design a variety of rate control algorithms.

3.1.1 The Principal Method

A defined section of the incoming bitstream is stored and analyzed in an input buffer. The size of the section can be interpreted as an analysis window. As a result of the analysis procedure, the requantization factors are determined. Then, the actual transcoding with the determined requantization factors is carried out. Obviously, the size of the analysis window affects the memory requirements and the delay of the deduced rate control algorithm. A rate control algorithm with low memory requirements and low delay uses a small analysis window. Therefore, in the CBR to CBR transcoding mode, the FDTC works on a slice-by-slice basis.

3.1.2 The Proposed Rate Control Algorithm

The FDTC stores all bits of one slice in an input buffer. The analysis procedure evaluates the quantization factor and the rate of each macroblock within the slice. Then, target rates for the entire slice and each macroblock are calculated by a bit allocation algorithm.

We define the complexity c of a macroblock after quantization approximately as

$$c = q \cdot d \qquad (1)$$

where q denotes the quantiser_scale and d denotes the rate of the macroblock. Not only the bit allocation algorithm is based on the complexity measure but also the rate control. Under the assumption that the complexity remains constant when varying the quantization factor and the rate, the reference requantization factor q_{out}^{ref} of a macroblock is given by

$$q_{out}^{ref} = \frac{q_{in} \cdot d_{in}}{d_{out}^{t}} \qquad (2)$$

where q_{in} and d_{in} denote the quantiser_scale and the rate of the macroblock in the input stream, respectively, and d_{out}^{t} is the target rate. The final requantization factor q_{out} is a weighted sum of q_{out}^{ref} and q_{in}.

After transcoding one slice the number of bits of the transcoded slice is compared with the target rate for the slice. The deviation is carried over to the next slice as a benefit or penalty to match the desired bit rate.

3.2 VBR to CBR Transcoding

3.2.1 Principle

The bit rate of a VBR bitstream varies with time. If such a bitstream is transmitted through a channel, the utilized bandwidth is a function of time. In contrast to a VBR bitstream, a CBR bitstream always allocates the same bandwidth. The bit rate characteristic reflects the amount of bits per picture. A VBR encoder can allow high peak output bit rates, e.g. for fast moving scenes with many details, whereas a CBR encoder has to cut peaks by using coarser quantization or other means. As a consequence, in a VBR bitstream the number of bits per picture varies much stronger with time than in a CBR bitstream. With CBR coding, the bit budget must be more equally distributed among all pictures. Eventually, complex scenes suffer from a lack of bits whereas for low activity scenes bits may be wasted. As a result, VBR coding normally supplies a better and often almost constant picture quality than CBR coding. These properties have to be taken into account when dealing with VBR to CBR transcoding. In order to get a CBR bitstream a picture with a great number of bits has to be requantized coarser than a picture with a small number of bits.

The principal method outlined in section 3.1.1 can also be used for a VBR to CBR rate control algorithm. Hence, a straightforward rate control procedure stores the bits of several pictures in advance, e.g. all pictures of a group of pictures, and computes the rate of each picture. This is necessary because of the strong variability of the rate in VBR bitstreams. For all stored pictures the requantization factors are determined in such a way as to output a CBR bitstream. Then, with a delay of several pictures, transcoding takes place. Obviously, a relatively large delay is introduced by this process and memory cost is increased although coded bits are stored merely. Hence, for low delay applications a different rate control method has to be found which is described in the following.

3.2.2 Low Delay VBR to CBR Transcoding

The VBR to CBR rate control algorithm of the FDTC neither explores the rate nor stores parts of the input bitstream. Most parameters of the rate control are deduced from the bitstream already transcoded. Reasonable assumptions on the basis of a priori knowledge are made for that parameters which are unknown due to the missing pre-analysis. The rate control algorithm is based on TM5 [4] and changes are made to comply with the requirements of VBR to CBR transcoding.

The first step comprises bit allocation. Before transcoding a picture, the target number of bits for the picture is estimated. This number depends on the target bit rate of the transcoded bitstream, the number of bits generated by transcoding the preceding pictures, the requantization factors of the preceding pictures, and the assumed group of pictures structure.

In the second step, the requantization factors are determined. The target number of bits estimated in step one is equally distributed among all macroblocks in the picture. However, the number of bits generated by transcoding the macroblocks of the picture differs from the assumed equal distribution. The deviation from equal distribution is used to calculate the requantization factor of the macroblock. This procedure works on a macroblock-by-macroblock basis. A transcoded macroblock is written into the output stream immediately after it is requantized. Then, the next macroblock is read from the input stream for requantization. Thus, the delay is negligible.

The rate control algorithm explained up to now is not able to handle VBR input bitstreams generally. The particular features of VBR bitstreams outlined in section 3.2.1 have to be considered more precisely.

The strong variability of the number of bits per picture in the input stream may lead to an improper requantization factor in step two. In order to produce a CBR output stream, the requantization factors in a picture with great activity are increased further to prevent a VBV buffer underflow. For a description of the VBV buffer, see [3]. On the other hand, stuffing bytes will be included in the transcoded bitstream, if the picture activity is too low, e.g. in black pictures or still pictures, and not enough bits can be generated by transcoding even with a small requantization factor.

Table I.

MEAN PSNR OF THE LUMINANCE COMPONENT FOR CBR TO CBR TRANSCODING. THE RESOLUTION OF THE TEST SEQUENCES WAS 720 X 576 PIXELS. ORIGINAL BITSTREAM DENOTES A DIRECTLY ENCODED BITSTREAM OF THE SPECIFIED BIT RATE.

Sequence	PSNR [dB]						
	Original Bitstream			Transcoded Bitstream 6 to 3 Mbit/s		Transcoded Bitstream 4 to 3 Mbit/s	
	6 Mbit/s	4 Mbit/s	3 Mbit/s	General Transcoder	FDTC	General Transcoder	FDTC
susie	43.12	42.13	41.31	40.35	40.39	40.05	40.35
football	35.13	32.89	31.21	30.71	30.39	30.33	30.79
mobile & calendar	30.21	28.01	26.42	25.65	25.73	25.53	26.23

The bit allocation in step one assumes a predefined group of pictures structure. Although many bitstreams use the same group of pictures structure, the rate control has to verifiy the assumption. If the assumption is invalid, a bit re-allocation will be done to reflect the new structure.

The rate control algorithm handles scene cuts. Scene cuts are detected by evaluating the macroblock types and the number of bits of the input bitstream. Especially in VBR bitstreams, the first picture after a scene cut is coded with many bits. The rate control chooses coarser requantization factors in order to create a CBR output stream.

4 Results and Discussion

4.1 CBR to CBR Transcoding

The FDTC with the rate control presented in section 3.1 was implemented and various MPEG-2 video bitstreams were applied to it. We used input bitstreams with 6 and 4 Mbit/s. They were transcoded to 3 Mbit/s. The resulting peak signal-to-noise ratio (PSNR) was compared with the output of the General Transcoder shown in Fig. 2. The results are listed in **Table I**. For comparison, the PSNR values of the original bitstreams directly encoded at 6, 4 and 3 Mbit/s are included. As can be seen in Table I, the FDTC achieves approximately the same PSNR values as the General Transcoder, although complexity and memory requirements are much lower. It is not surprising that transcoding to a lower bit rate results in a lower PSNR compared to direct encoding of the source material at the lower bit rate. **Fig. 4** shows in detail the PSNR of a bitstream transcoded by the General Transcoder and the FDTC. The CBR characteristic of the transcoded bitstreams was verified by regarding the VBV buffer occupancy.

4.2 VBR to CBR Transcoding

The performance of the VBR to CBR transcoding algorithm was checked with various VBR bitstreams comprising scene cuts, pictures with low activity, fade in, fade out, and varying group of pictures structures.

Fig. 5 exemplarily shows the performance of the FDTC in case of pictures with an extremely low activity. The VBR input stream was created by including nine black pictures in the football sequence. The first stage in the MPEG-2 decoder at the receiver is a VBV buffer specified in [3]. The decoder takes bits out of this buffer on a picture-by-picture basis and is fed by the CBR bitstream of the FDTC. The left diagram in Fig. 5 depicts the VBV buffer occupancy. Note, as the coded black pictures contain only very few bits, the VBV buffer occupancy increases with time. To prevent this buffer from overflow, stuffing bytes simulating picture content have to be sent by the FDTC after the first few black pictures. As a consequence, the buffer occupancy remains on its maximum level. The included stuffing bytes can also be seen in the right diagram in Fig. 5. In the transcoded bitstream, the number of bits is generally lower than in the input stream. However, this is not true when stuffing bytes are included due to VBR to CBR transcoding.

5 Implementation on a Media Processor

The FDTC was implemented in real time on a media processor [5]. The execution time depends on the chosen output bit rate. For example, transcod-

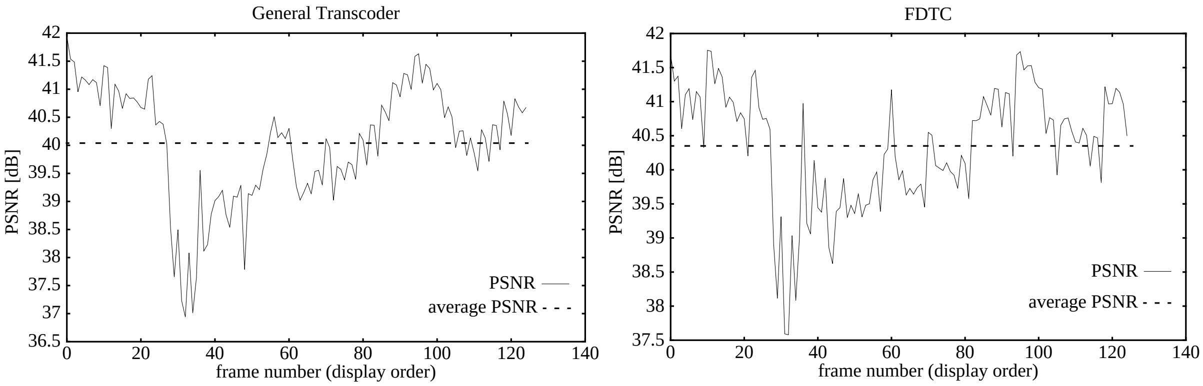

Fig. 4. Frame-by-frame PSNR comparison of the General Transcoder (left) and the FDTC (right) of the luminance component for CBR to CBR transcoding. The test sequence susie was transcoded from 4 to 3 Mbit/s.

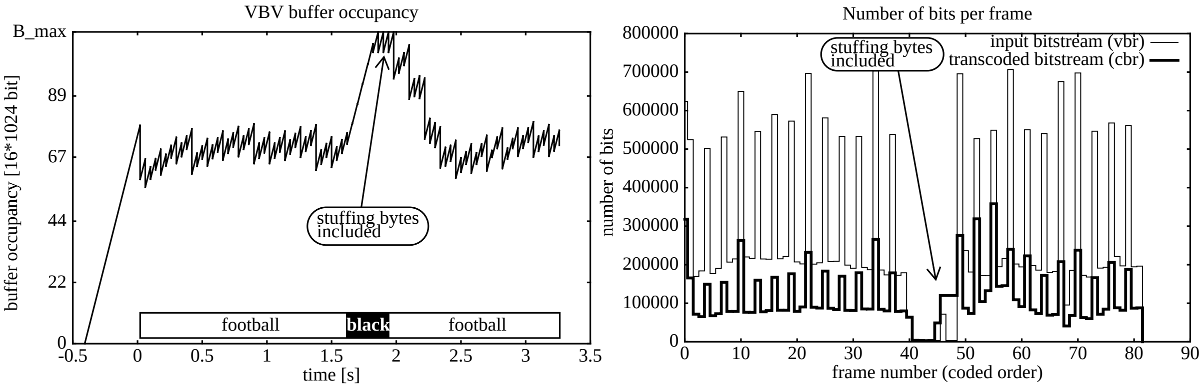

Fig. 5. VBR to CBR transcoding. A VBR input bitstream with some black pictures is transcoded to a CBR bitstream. Stuffing bytes are included to prevent VBV buffer overflow. The average bit rate of the VBR input stream is about 7 Mbit/s. The transcoded CBR bitstream has a bit rate of 3 Mbit/s.

ing the five seconds long test sequence susie from 6 to 3 Mbit/s takes about 3.5 seconds.

6 Conclusion

The presented FDTC is a MPEG-2 video transcoder with low complexity and low memory requirements. The implemented rate control is able to handle both VBR and CBR input bitstreams and produces a CBR stream at the output. Since the rate control operates on a slice-by-slice or macroblock-by-macroblock basis, the delay is low.

The FDTC achieves the same PSNR as the General Transcoder which is a cascade of a decoder and an encoder. Moreover, the FDTC can be implemented in real time using a modern media processor. Thus, it is a useful system to adapt MPEG-2 video bitstreams to the lower bit rate of transmission channels and/or storage systems.

Acknowledgement

The authors are indebted to the Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg and XSYS, Villingen-Schwenningen, for the support.

References

[1] B. Sostawa, Internal project report “I&K im Automobil” funded by the Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg.

[2] Project communication with Dr. D. Teichner, XSYS, and his group.

[3] ISO/IEC 13818-2, “Information technology – Generic coding of moving pictures and associated audio information: Video”, 1996.

[4] Test Model 5, Draft Revision 2, April 1993.

[5] Philips TriMedia Product Group, “TM1000 Preliminary Data Book”, 1997.