

Transcoding of MPEG bitstreams

Gertjan Keesman^{a,*}, Robert Hellinghuizen^b, Fokke Hoeksema^b, Geert Heideman^b

^a Philips Research Laboratories Eindhoven, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

^b University of Twente, TIOS Group, P.O. Box 217, 7500 AE Enschede, The Netherlands

Received 25 May 1995

Abstract

This paper discusses the problem of transcoding as it may occur in, for instance, the following situation. Suppose a satellite transmits an MPEG-compressed video signal at say 9 Mbit/s. This signal must be relayed at a cable head end. However, since the cable capacity is only limited, the cable head end will want to relay this incoming signal at a lower bit-rate of, say, 5 Mbit/s. The problem is how to convert a compressed video signal of a given bit-rate into a compressed video signal of a lower bit-rate. The specific transcoding problem discussed in this paper is referred to as *bit-rate conversion*. Basically, a transcoder used for such a purpose will consist of a cascaded decoder and encoder. It is shown in the paper that the complexity of this combination can be significantly reduced. The paper also investigates the loss of picture quality that may be expected when a transcoder is in the transmission chain. The loss of quality as compared to that resulting in the case of transmission without a transcoder is studied by means of computations using simplified models of the transmission chains and by means of using computer simulations of the complete transmission chain. It will be shown that the presence of two quantizers, i.e. cascaded quantization, in the transmission chain is the main cause of extra losses, and it will be shown that the losses in terms of SNR will be some 0.5–1.0 dB greater than in the case of a transmission chain without a transcoder.

Keywords: Video compression; MPEG; Transcoding

1. Introduction

The use of digital transmission formats is proliferating fast. The connection of several transmission media may give rise to the following problem. Suppose a program provider transmits, e.g. across a satellite link, a digital video signal in a compressed format and this program is to be relayed e.g. on a cable network. The relayed signal must

also be in the compressed format. Now, suppose the bit-rate of the digital video signal on the satellite link is R_1 (Mbits/s), and suppose that the cable network has a limited capacity, as a result of which the incoming program can only be relayed if its bit-rate is R_2 (Mbit/s) or less. The problem is that the incoming compressed signal of one bit-rate has to be converted into a compressed signal of a lower bit-rate. The operation of converting a compressed format into another compressed format is called *transcoding* and a device that performs this conversion is called a *transcoder*. In this paper the specific transcoding problem of bit-rate conversion is

* Corresponding author. Tel.: (+31)40 2743721; fax: (+31)40 2744675; e-mail: keesman@natlab.research.philips.com.

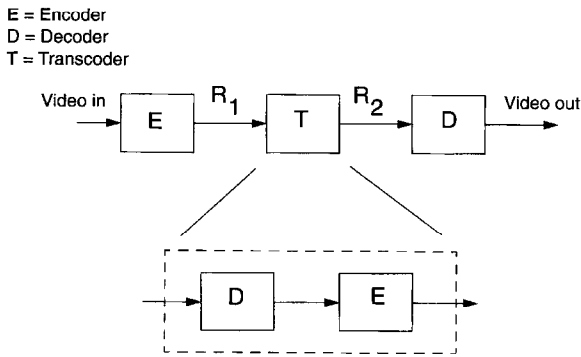


Fig. 1. Basic configuration of a system including a transcoder.

discussed. Since MPEG video compression (see e.g. [2, 3, 5, 6, 8, 10, 12]) is the current mainstream in compression technology this paper will concentrate on the transcoding of MPEG signals into MPEG signals.

Basically, the situation with transcoding is as shown in Fig. 1: an encoder compresses the incoming video signal at a bit-rate of R_1 (Mbit/s), then this compressed signal is converted into a compressed format of a lower bit-rate R_2 (Mbit/s) and finally a decoder decompresses the incoming signal and displays the resulting video signal. Transcoding may occur in situations where one means of signal transport interfaces another means of signal transport.

Essentially, a transcoder consists of a cascaded decoder and encoder, as shown in the lower part of Fig. 1. In the situation described in this paper the encoder embedded in the transcoder is only compressing at a lower bit-rate than the incoming bit-rate, so there is no other reformatting involved, such as resampling.

In the literature the problem of transcoding seems fairly uncovered. In [9] the problem of PCM-ADPCM-PCM lossless transcoding is discussed. Morrison et al. [7] describe a transcoder that has a limited complexity for video signals compressed with hybrid DCT systems. Morrison et al. compare the picture quality in a situation when transcoding from bit-rate R_1 to R_2 with the situation where the video signal is directly compressed at R_2 . Their measurements indicate an extra loss

of approximately 1 dB for H.261 compressed signals.

Transcoding by means of cascading a decoder and an encoder may lead to several types of problems. In this paper two problems are described. The first problem is that of complexity. It will be shown, in a way similar to that of Morrison [4], that the complexity can be reduced significantly. The transcoder discussed in this paper is, however, slightly less complex than that of Morrison. Another difference with respect to Morrison et al. is that the transcoder discussed in this paper will be applied to MPEG-compressed signals instead of H.261-compressed signals as in the case of Morrison.

The second problem is that of performance, that is, picture quality at a given bit-rate. The introduction of a transcoder in the transmission chain may introduce extra losses as compared to compressing directly, called *direct coding*, to the lowest bit-rate. This can be investigated via two approaches: (1) theoretical analysis and (2) computer simulation. Signals passing through transcoders are typically quantized twice. This cascaded quantization is a source of performance loss in the case of trans-coded signals. Besides theoretically analyzing cascaded quantization it is also possible to simulate a complete transmission path including a transcoder. The latter does not involve any model abstraction, but may yield less insight into the mechanisms which actually cause performance loss.

This paper is organized as follows. Section 2 will discuss the problem of reducing the complexity of the transcoder. The problem of picture quality loss will be discussed in the following three sections. A typical source of performance loss in a system including transcoding is the double occurrence of a quantization operation. The first step is to analyze cascaded quantization straightforwardly, which will be discussed in Section 3. This section will present an analysis for single coefficients so that insight into the basic mechanisms can be obtained and this section will present an analysis for the combination of 64 coefficients. Section 3.4 will discuss the second step, which is to enhance the quantization model with prediction loops. Results obtained in computer simulations of a system including transcoding will be discussed in Section 4. Finally, the paper will be concluded in Section 5.

2. Complexity of transcoder

In the introduction a transcoder was pictured as a cascaded decoder and encoder. Straightforward cascading of the decoder and encoder, however, may lead to far too complex transcoders. In this section it will be shown that the transcoder can be much simpler, i.e. the number of (I)DCTs and the amount of memory required can be reduced by combining the decoder and encoder in a more efficient way. The approach will be to cascade a decoder and an encoder and then analyze which elements of the two can be combined.

The transmission chain being studied comprises two cascaded encoding and decoding operations. The variables associated with the first encoding and decoding operation will have superscript (1) and the variable associated with the second encoding and decoding operation will have superscript (2). In system blocks subscripts 1 and 2 will be used to denote this difference.

Fig. 2 shows the outline of a decoder that is present in the transcoder. First the bit-stream is decoded (VLD), yielding the value of the quantized coefficients, next these quantized coefficients are de-quantized (DQ₁) and put through an inverse discrete cosine transform (IDCT). This process yields the residual signal $R_n^{(1)}$ of the picture $I_n^{(1)}$. The picture can be reconstructed by adding the prediction which results from the previously decoded picture $I_{n-1}^{(1)}$ by applying motion compensation (shift).

Let V_n denote the vector field of picture n . In fact, each component of the vector represents a displacement

for a pixel. If the position of a pixel in a picture is denoted by \mathbf{x} then the displacement is here denoted $V_n(\mathbf{x})$. The motion-compensated signal is denoted $S[I_{n-1}^{(1)}, V_n]$ in Fig. 2 and the operation $S[\cdot]$ will be referred to as the shift operation. The motion compensation yields the prediction $P_n = S[I_{n-1}^{(1)}, V_n]$ as follows:

$$P_n(\mathbf{x}) = S[I_{n-1}^{(1)}, V_n](\mathbf{x}) \\ = I_{n-1}^{(1)}(\mathbf{x} + V_n(\mathbf{x})).$$

According to the MPEG syntax, the displacements $V_n(\mathbf{x})$ for each pixel are equal for pixels within a macroblock. The macro-block vectors are transmitted in the bit-stream. So, besides quantized DCT coefficients, motion vectors are also decoded from the incoming bit-stream. Furthermore, the so-called macro-block modes are also decoded from the incoming bit-streams. These macro-block modes contain information about, for instance, the intra/inter decision or the forward/backward/interpolated decision [3, 10].

A significant reduction in complexity can be achieved if the cascaded encoder uses the same picture types as the incoming signal. So, when a decoded I picture is again coded as an I picture, a decoded P picture is again coded as a P picture and a decoded B picture is again coded as a B picture, the transcoder complexity can be reduced due to two factors.

First of all, the decoded pictures are not yet in the proper order. That is, although B pictures are decoded after the corresponding anchor frames, they are displayed between the two anchor frames. This

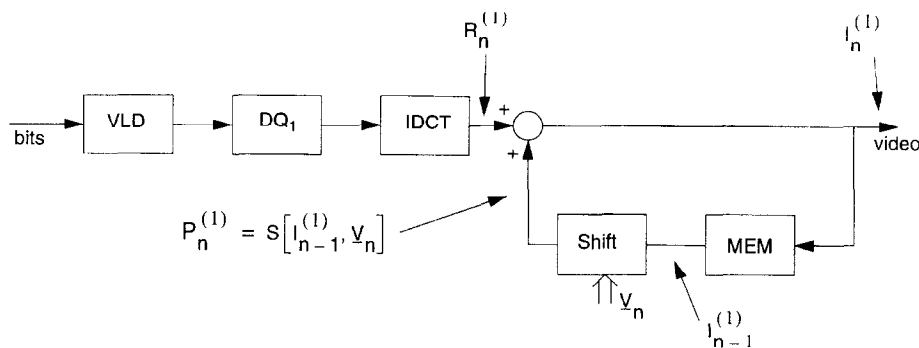


Fig. 2. Outline of the decoder in the transcoder.

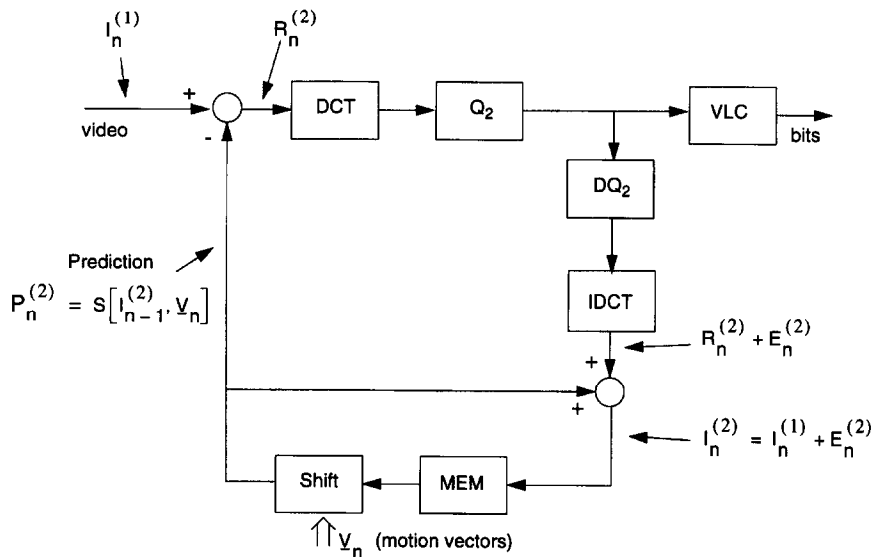


Fig. 3. Outline of the encoder in the transcoder.

so-called picture reordering is performed inversely in the decoder. The picture reordering in the decoder and the picture ordering in the encoder cancel out when the two are cascaded. Here a large amount of memory can be saved.

Secondly, most of the decoded information, such as motion vectors and macro-block modes, can be reused in the cascaded encoder. Especially, the obviated need to estimate motion vectors again reduces the computational complexity of the transcoder significantly.

At this point there is no need to have decoded pictures available in the transcoder because most of the incoming data can be copied from the decoder to the encoder. In the following it will be shown that, because there is no need for decoded pictures, the amount of memory for storing the previously decoded pictures can also be reduced.

The typical MPEG-2 picture level features (see e.g. [3]) can also be copied from the incoming bit-stream to the encoder. This holds for features such as 'picture-structure', 'top-field-first', 'repeat-first-field' and 'progressive-frame'.

Fig. 3 shows the basic outline of the encoder. First the prediction $P_n^{(2)}$ is subtracted from the incoming pictures $I_n^{(1)}$, which yields the residual signal $R_n^{(2)}$. This residual signal is put through a discrete cosine transform (DCT) and a quantizer (Q_2).

The quantized DCT coefficients are compressed and then outputted. Also, the quantized DCT coefficients are again dequantized and are further fed through an IDCT to yield the reconstructed residual picture $R_n^{(2)} + E_n^{(2)}$, where $E_n^{(2)}$ denotes the quantization error in the second encoder, i.e. the encoder in the transcoder.

In the next step the picture $I_n^{(2)}$ is reconstructed by adding the same prediction as was subtracted earlier in the process. This prediction is obtained by applying the shift operation $S[\cdot]$ to the previously decoded picture.

The basic transcoder consisting of a decoder (see Fig. 2) and a cascaded encoder (see Fig. 3) is shown in Fig. 4. Observe that only the prediction loop of the decoder is shown. The transcoder of Fig. 4 can be simplified by removing one of the two picture stores (MEM).

This can be done using the fact that the shift operator is linear, that is,

$$S[I_{n-1}^{(1)} + E_{n-1}^{(2)}, V_n] = S[I_{n-1}^{(1)}, V_n] + S[E_{n-1}^{(2)}, V_n] \quad (1)$$

holds. Using this linearity property the prediction in the second encoder, i.e. the encoder within the

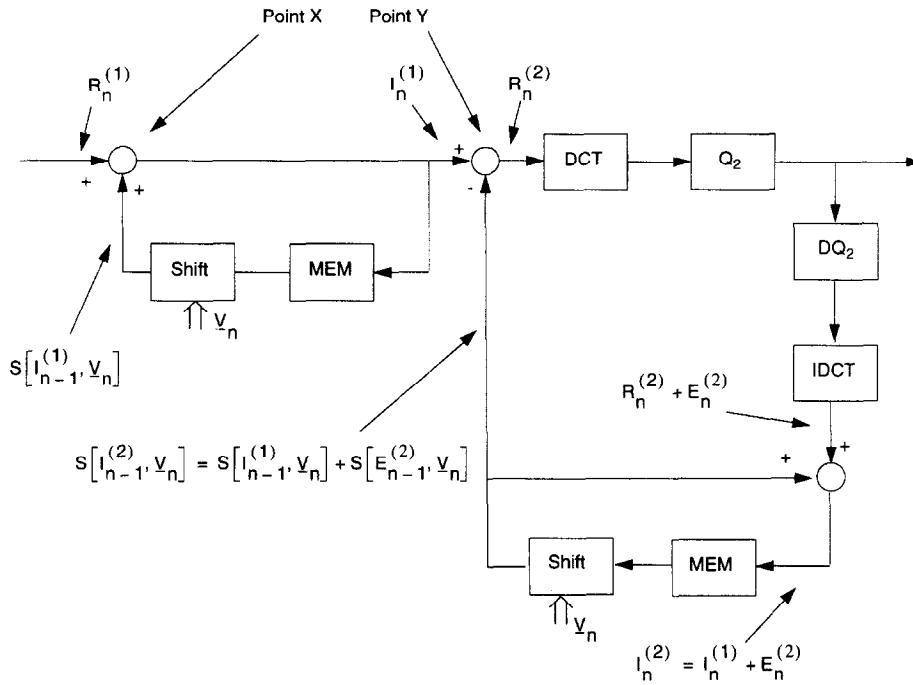


Fig. 4. Cascaded decoder and encoder as transcoder.

transcoder, can be split into two components. One of the two components in the encoder prediction results from the motion compensation applied to the previously decoded picture $I_{n-1}^{(1)}$. Observe that this prediction is first added in the decoder at ‘point X’ in the figure and is then subtracted in the second encoder at ‘point Y’ in the figure. These two operations of course cancel one another. It is only necessary to subtract the motion-compensated quantization error, as will be shown in the following.

Using Eq. (1), the residual $R_n^{(2)}$ in the second encoder can be written as

$$R_n^{(2)} = I_n^{(1)} - S[I_{n-1}^{(1)} + E_n^{(2)}, V_n] \\ = I_n^{(1)} - S[I_{n-1}^{(1)}, V_n] - S[E_{n-1}^{(2)}, V_n]. \quad (2)$$

The decoded picture $I_n^{(1)}$ can be written as

$$I_n^{(1)} = R_n^{(1)} + S[I_{n-1}^{(1)}, V_n]. \quad (3)$$

Substitution of Eq. (3) into Eq. (2) yields

$$R_n^{(2)} = R_n^{(1)} - S[E_{n-1}^{(2)}, V_n]. \quad (4)$$

All that is needed to be able to produce the outgoing bit-stream is the residual $R_n^{(2)}$. According to Eq. (4), this residual $R_n^{(2)}$ can be obtained directly by subtracting the motion-compensated quantization error $E_{n-1}^{(2)}$. Hence, only $E_n^{(2)}$ need be stored, and the store used for $I_n^{(1)}$ can be dropped.

Fig. 5 shows the block diagram resulting after the removal of a picture store. To obtain the coding error $E_n^{(2)}$ the input and the output of the coding path consisting of DCT, quantizer, de-quantizer and IDCT are subtracted. Consecutively the coding error $E_n^{(2)}$ is stored in a memory. The residual signal $R_n^{(2)}$ is obtained by subtracting the motion-compensated error $S[E_{n-1}^{(2)}, V_n]$ from the decoded residual $R_n^{(1)}$.

Another simplification can be realized in the transcoder of Fig. 5 by moving the DCTs and IDCTs around in the block diagram. The DCTs and IDCTs can be moved around by using the linearity property of the DCT. Fig. 6 shows some of the possibilities. The top section of the figure shows that in a data fork the DCT (or IDCT) can be moved from the main path to the two separating

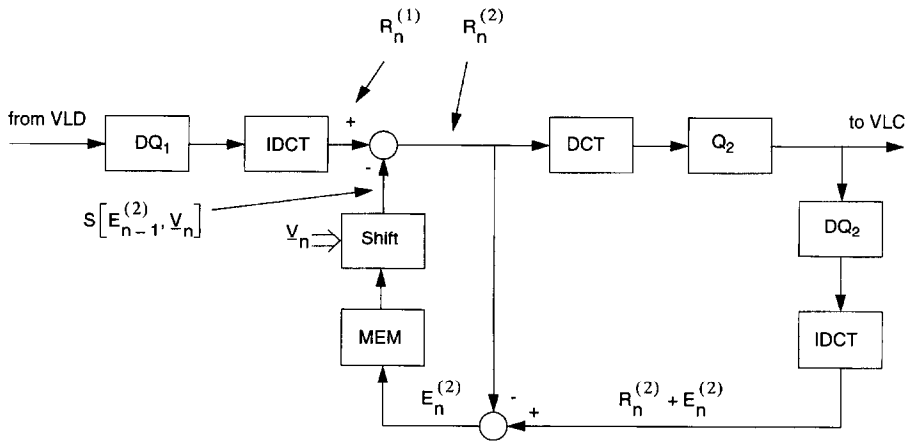


Fig. 5. Transcoder with a reduced amount of memories.

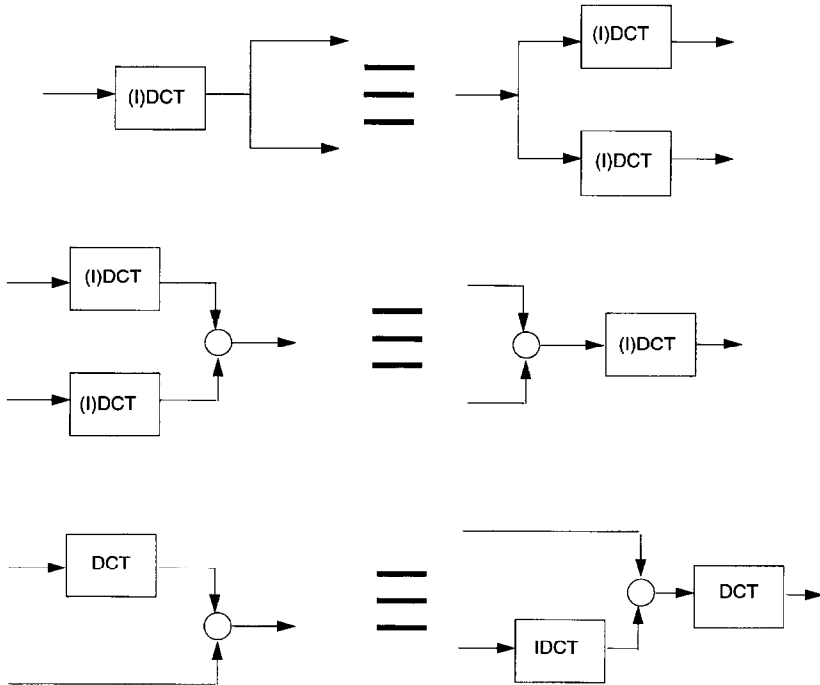


Fig. 6. Moving DCT and IDCT operations around.

paths; the middle section shows that at an addition or a subtraction the DCT (or IDCT) of the components of the addition can be moved to the DCT (or IDCT) of the result. Finally, applying an IDCT to a signal that has passed through a DCT equals a unity transfer. This property is used in the lower

part of Fig. 6 in combination with the property for an addition.

Moving the DCTs and IDCTs of Fig. 5 around in this way yields the transcoder of Fig. 7. Observe that the number of DCTs and IDCTs has been reduced from three to two. At this point the com-

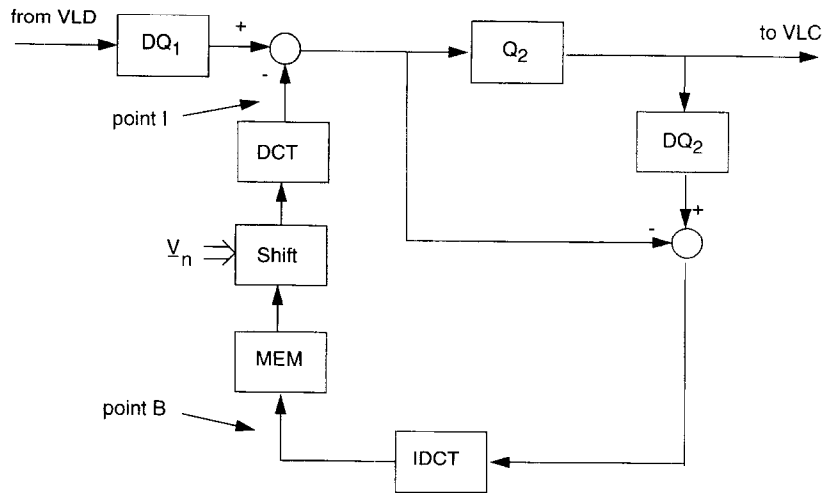


Fig. 7. Transcoding system with a reduced number of DCTs.

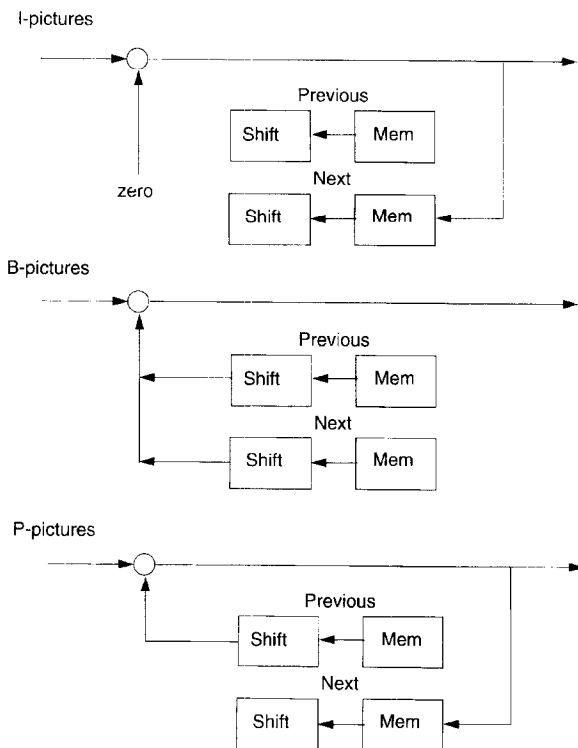


Fig. 8. Data flow in decoder for I, B and P pictures.

plexity of the transcoder is that of a decoder, plus an extra DCT, an extra quantizer, de-quantizer and a VLC with a buffer.

The whole procedure followed for deriving the transcoder of Fig. 7 is implicitly based on the use of P pictures. The derivation is, however, equally valid for I and B pictures, which will be shown next.

Fig. 8 shows the decoder prediction loop for I, B and P pictures. When an I picture is decoded, the prediction equals zero and the decoded picture is stored in one of the anchor frame memories. Replacing the prediction by zero does not affect its linearity, and the whole derivation remains valid. In Fig. 7 a switch has to be introduced as 'point I' in the figure for an I picture. So, in the case of an I picture, the prediction subtracted from the decoded residual signal equals zero. The quantization error still has to be computed and stored as usual because the I picture is used as an anchor frame, that is, it is used to predict future pictures. Observe that in the case an open switch at 'point I' the transcoder is simply dequantizing the picture and again quantizing the picture, that is, there is only a cascaded quantization.

The middle part of Fig. 8 shows the configuration when B pictures are decoded. In this case the prediction is formed from both the previous and the next picture by means of interpolation. The interpolation does not change the linearity of the shift operation and, hence, the shown derivation holds for B pictures too. Because the B pictures are not used for any further pictures there is no need to

Table 1

Cascaded dec. & enc.	Simplified transcoder
3 (I)DCTs	2 (I)DCTs
2 Anchor stores	1 Anchor store
3 Adds	2 Adds

store them. This is indicated in Fig. 7 by ‘point B’, indicating that in the case B pictures a switch at this point is opened so that the quantization errors for B pictures are not stored.

The lower part of Fig. 8 shows the configuration for P pictures. It has been amended by putting switches on two memories, storing both the previous and next anchor frames. Again, the linearity of the shift operation is not affected and hence the derivation remains valid.

This section can be concluded as follows. A transcoder can have a significantly reduced complexity as compared with a cascaded decoder and encoder. Table 1 shows a comparison between the straight-forward cascaded decoder and encoder and the transcoder derived here.

It is assumed that in both solutions for transcoding, the macro-block modes and motion vectors are obtained from the incoming bit-stream. The next problem is how much of performance loss does the introduction of a transcoder in the transmission chain cause?

3. Analysis of transcoder performance

Transmitting a video signal across a system as shown in Fig. 1 may lead to extra distortion as compared to directly compressing the video signal at the final bit-rate R_2 (Mbit/s). In this section a first step is made in analyzing the performance of transmission across a transcoder.

The transmission chain model of Fig. 1 is shown in greater detail in Fig. 9. In fact, Fig. 9 is obtained by substituting Fig. 3 for the encoder ‘E’, Fig. 7 for the transcoder ‘T’ and Fig. 2 for the decoder ‘D’. When coding I pictures the prediction signals that are being subtracted/added are all set to zero. So in this case the pictures are put through the DCT, quantized, dequantized, quantized, dequantized and finally put through an IDCT. Characteristic is the double quantization, which is considered to be the mechanism that introduces extra distortion in a system involving transcoding.

This section discusses cascaded quantization and its effects. Section 3.1 discusses the quantization model that will be studied. Two cascaded quantizers can be replaced by a single *effective quantizer*. Section 3.2 defines this effective quantizer and Section 3.3 describes the performance of this effective quantizer both for individual coefficients and for a combination of 64 coefficients.

Computational convention. It will be assumed that the quantizers under consideration are symmetrical at positive and negative values. The quan-

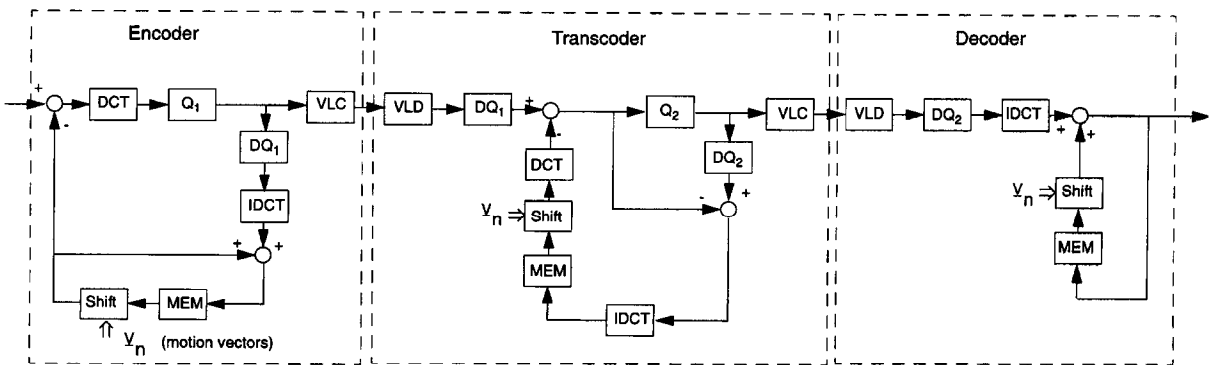


Fig. 9. Transmission chain with transcoder.

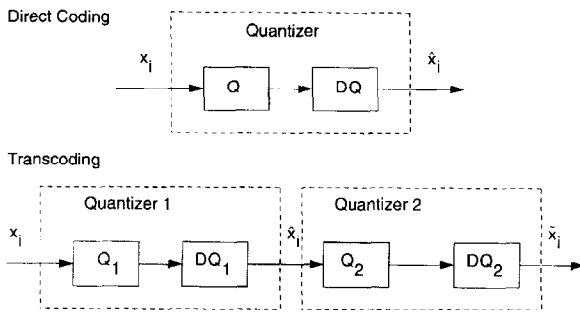


Fig. 10. Quantization model reflecting the rudimentary transcoding system.

tizer will be specified for positive values; this can easily be adapted for negative values.

3.1. Quantization model

Fig. 10 shows the two quantizer models that will be compared. The top part of the figure shows the reference system, which is assumed to reflect direct coding; the bottom part of the figure shows the system under consideration, that is, two cascaded quantizers. The two quantized quantizers can also be seen in Fig. 9, in the main path.

It will be assumed that there are two complementary operations: *quantization* (Q) and *de-*

quantization (DQ). A *quantizer* involves a quantization and a de-quantization operation.

In Fig. 10 quantization is denoted by ‘Q’ and refers to the process of mapping an incoming real-valued sample on a discrete number $n \in [-N, +N]$, that is,

$$Q: \mathbb{R} \rightarrow \{-N, \dots, 0, \dots, +N\}.$$

In the case of quantization the axis with the real-valued numbers is partitioned into $2N + 1$ disjunct quantization intervals θ_i . An incoming sample x is mapped on the discrete number n if x falls in the interval θ_n .

Usually, the intervals θ_n are represented by their *decision thresholds* t_n . Here it is assumed that for $n > 0$ the intervals θ_n equal

$$\theta_n = (t_{n-1}, t_n],$$

for $n = 0$ the interval θ_0 equals

$$\theta_n = [-t_0, t_0],$$

and for $n < 0$ the interval θ_n equal

$$\theta_n = [-t_{-n}, -t_{-n-1}).$$

As mentioned above, it is assumed that the quantizers are symmetrical around zero.

The top part of Fig. 11 illustrates the quantization operation. The horizontal line in the figure represents the set \mathbb{R} . For instance, if an incoming

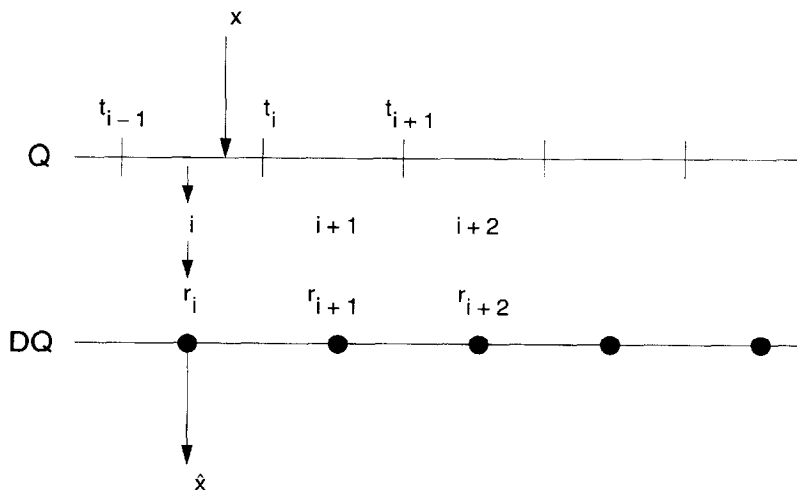


Fig. 11. Quantization and de-quantization processes.

sample with a value x falls in interval θ_i , which is bounded by the decision threshold t_{i-1} and t_i , it is mapped on $Q(x) = i$.

In Fig. 10 de-quantization is denoted by ‘DQ’ and refers to the process of mapping an incoming discrete number $n \in [-N, +N]$ on a real-valued number $\hat{x} \in \mathbb{R}$, that is,

$$\text{DQ}: \{-N, \dots, 0, \dots, +N\} \rightarrow \mathbb{R}.$$

In the case of de-quantization each incoming discrete number is mapped on a real-valued number $\hat{x} = r_n$. These real-valued numbers r_n are referred to as the *representation levels* of the de-quantizer.

In the case of the MPEG intra-quantizer the representation levels are equidistant, that is, let Δ denote the step size, then the representation levels equal

$$r_n = n \Delta. \tag{5}$$

In MPEG systems the value of the step size Δ can be decoded from the quantization matrix and the value of ‘mquant’ which is transmitted in the bit-stream.

If optimum performance is desired the two steps, quantization and de-quantization, have to be designed jointly. General procedures for doing this are, for instance, ‘Max-Lloyd quantization’ or ‘entropy-constrained quantization’ [4]. In the case of MPEG systems, the de-quantizer is specified in the standard and this leaves no room for specific designs. The quantizer, however, is not defined by the MPEG standard and a number of possibilities are still open. In the following analysis it will be assumed that the decision thresholds are at 5/8 between two representation levels, that is, for $n \geq 0$,

$$t_n = \Delta \left(n + \frac{5}{8} \right) \tag{6}$$

holds. This quantizer is used in, for instance, MPEG-2 TM5 [8].

The position of the representation levels of the MPEG quantizers has been standardized. In MPEG it is possible to design ‘optimal’ quantizers in the following sense. Given the statistical properties of the DCT-coefficients and given the required bit-rate for a picture, it is possible to compute the decision levels of the quantizer in such a way that

the resulting distortion is minimal. In fact, this is an entropy-constrained [1] quantizer that has been further constrained in the possible choices of its representation levels. The TM5 quantizer specified by Eq. (6) leads to a more regular spacing of the decision levels which will give it an easier implementation, while its performance is very close to that of ‘optimal quantizer’. In the following it will be used for intra-quantizers and it will be referred to as the TM5 quantizer.

3.2. Effective quantizer

The cascaded quantizer of Fig. 10 can be replaced by one single *effective quantizer*, which is illustrated in Fig. 12. The first quantizer maps incoming sample values x on a quantized version \hat{x} . These quantized values \hat{x} are again quantized and mapped on the values \tilde{x} .

Let Δ_1 denote the step size of the first quantizer and let Δ_2 denote the step size of the second quan-

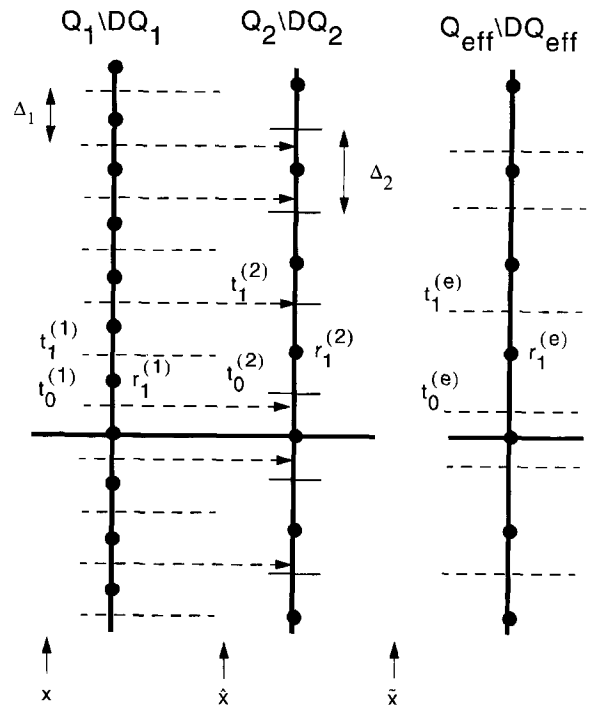


Fig. 12. Decision levels and representation level of the effective quantizer.

tizer. Variables with a superscript (e), e.g. $r_n^{(e)}$ and $t_m^{(e)}$, refer to the effective quantizer. Because the decision intervals of the effective quantizer will have an irregular size, there is no sense in denoting a variable for its step size.

The representation levels of this effective quantizer are simply the representation levels of the second quantizer. The decision levels are formed by the decision levels of the first quantizer in the following way. Take all the representation levels $r_n^{(1)}$ of the first quantizer that fall in a certain decision interval $\theta_m^{(2)}$ of the second quantizer. Next, take the union

$$\theta_m^{(e)} = \bigcup_{r_n^{(1)} \in \theta_m^{(2)}} \theta_n^{(1)} \quad (7)$$

of all the corresponding decision intervals $\theta_n^{(1)}$ of the first quantizer. This union forms the decision interval $\theta_m^{(e)}$ of the effective quantizer.

The decision thresholds of the effective quantizer can be obtained by computing the largest representation level $r_{n_{\max}}^{(1)}$ that falls in decision interval $\theta_m^{(2)}$, that is, n_{\max} has to be solved from

$$r_{n_{\max}}^{(1)} = \max_n \{r_n^{(1)} \mid r_n^{(1)} \in \theta_m^{(2)}\}.$$

The n_{\max} is of course a function of m .

Assuming TM5 quantizers for both the first and the second quantizer makes it possible to actually

compute these effective decision thresholds $t_m^{(e)}$. The problem is how to compute n_{\max} from

$$r_{n_{\max}}^{(1)} = \max_n [r_n^{(1)} < t_m^{(2)}],$$

with

$$t_m^{(2)} = \Delta_2 \left(m + \frac{5}{8}\right)$$

and

$$r_n^{(1)} = \Delta_1 n.$$

Solving this yields for n_{\max} :

$$n_{\max} = \left\lfloor \frac{\Delta_2}{\Delta_1} \left(m + \frac{5}{8}\right) \right\rfloor. \quad (8)$$

Using Eq. (6) after substitution of Δ_1 for Δ yields for the decision levels $t_m^{(e)}$ of the effective quantizer:

$$t_m^{(e)} = \Delta_1 \left(\left\lfloor \frac{\Delta_2}{\Delta_1} \left(m + \frac{5}{8}\right) \right\rfloor + \frac{5}{8} \right). \quad (9)$$

The representation levels of the effective quantizer are simply the representation levels of the second quantizer, that is,

$$r_m^{(e)} = \Delta_2 m. \quad (10)$$

Fig. 13 shows the decision levels and the decision levels of both direct quantization and cascaded

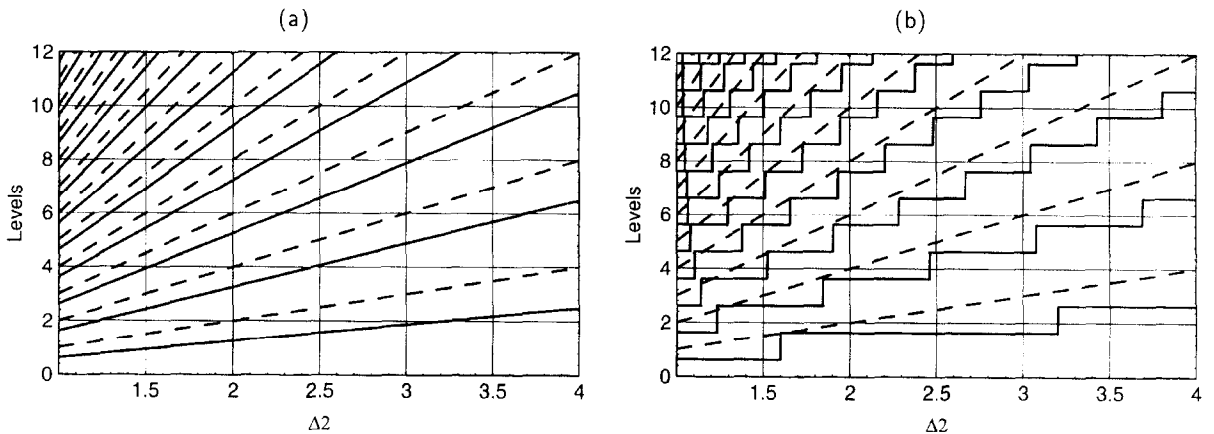


Fig. 13. Quantizer levels as functions of the step size Δ_2 . Solid lines represent decision thresholds and dashed lines represent representation levels and the line corresponds to increasing n . (a) Thresholds and representation levels in the case of direct quantization. (b) Thresholds and representation levels in the case of cascaded quantization, using $\Delta_1 = 1.0$.

quantization. The left graph shows the case of direct quantization: all solid lines represent decision thresholds

$$t_n = \Delta(n + \frac{5}{8}).$$

The lowest horizontal line represents $n = 0$ and every next solid line represents an n increased by one. Since the decision levels are shown as functions of Δ , the slopes of the lines increase with an increasing n . The dashed lines represent the representation levels in a similar way.

The right part of Fig. 13 shows the representation levels and the decision thresholds for the cascaded quantization case, i.e. the effective quantizer, as a function of the step size Δ_2 of the second quantizer. For these curves the step size of the first quantizer is set to a fixed value of $\Delta_1 = 1.0$. Since the representation levels are simply the representation levels of the second quantizer, the dashed lines follow a similar pattern as in the left graph. However, the decision levels follow Eq. (9). The floor function causes jumps in the decision levels. If the step size of the second quantizer is increased, more and more representation levels $r_n^{(1)}$ will fall in a decision interval $\theta_i^{(2)}$. The jumps in the figure coincide with the step size Δ_2 at which a representation level of the first quantizer moves to a lower decision interval, that is, it no longer falls in interval $\theta_i^{(2)}$, but in $\theta_{i-1}^{(2)}$.

The jumps in the curve are at the points where the decision intervals change. The effective quantization operation remains the same between two jumps. Accordingly, the entropy of the output symbols only changes at these jumps. These jumps will of course result in a distortion-entropy curve which will be highly irregular, as will be shown in the following section.

3.3. Performance of cascaded quantization

In this section the performance in entropy rate sense of two cascaded quantizers will be computed and the results will be compared with the results obtained for direct quantization. Actually, this section will discuss the performance in the case of one DCT coefficient and in the case of combining 64 DCT coefficients. The first case corresponds to

a complete isolation of cascaded quantization which allows to investigate its effects, whereas the case of 64 DCT coefficients corresponds to the situation that occurs in coding I pictures.

The input to the cascaded quantizers are values of DCT coefficients. It will be assumed that these x values have a two-sided Laplace pdf [11], that is,

$$p(x) = \frac{1}{\sqrt{2}\sigma} e^{-|x|/\sqrt{2}\sigma}. \quad (11)$$

Using this, it is possible to compute characteristics such as distortion and entropy. The distortion of a quantizer is the difference between the input and output, that is $d = x - \hat{x}$. Quantizers are usually characterized by the variance σ_d^2 of the distortion, which is specified by

$$\begin{aligned} \sigma_d^2 &= E\{(x - \hat{x})^2\} \\ &= \int_{-\infty}^{\infty} (x - \hat{x})^2 p(x) dx \\ &= \sum_n \int_{\theta_n} (x - r_n)^2 p(x) dx. \end{aligned} \quad (12)$$

Often the distortion is given in a logarithmic format by means of the SNR. In image processing the following definition for SNR is used:

$$\text{SNR} = 10 \log_{10} \left(\frac{255^2}{\sigma_d^2} \right). \quad (13)$$

This definition for SNR will be used both for measuring the distortion in a single coefficient and for the actual SNR for a picture. In the latter case, i.e. the SNR for a picture, the variance σ_d^2 is replaced by the sum of the variances over all coefficients $\sum_i \sigma_{d_i}^2$. The value 255 refers to the maximum possible pixel value, that is, Eq. (13) actually refers to a peak-signal to noise ratio and consequently some authors refer to it as PSNR. In this paper, when the SNR is computed for a single coefficient it will be referred to as the *coefficient SNR*.

The entropy of the output of the quantization can also be computed using Eq. (11). The probability of x falling in decision interval θ_n is denoted by p_n , where

$$p_n = \int_{\theta_n} p(x) dx. \quad (14)$$

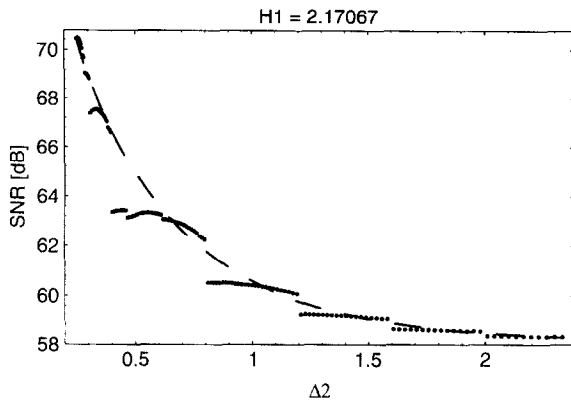


Fig. 14. SNR as a function of the step size for direct quantization (dashed line) and for cascaded quantization (solid line) with $\Delta_1 = 1.0$.

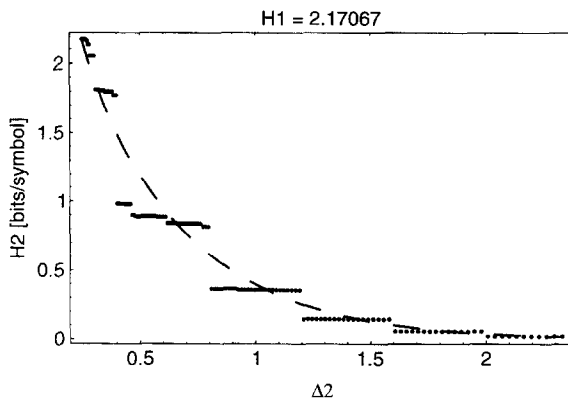


Fig. 15. The entropy of the quantizer output for direct quantization (dashed line) and for cascaded quantization (solid line) with $\Delta_1 = 1.0$ and $\sigma^2 = 1.0$.

So the entropy of the output signal of the quantizer equals

$$H = -\sum_n p_n \log_2 p_n \quad (\text{bits/symbol}). \quad (15)$$

Fig. 14 shows the coefficient SNR as a function of the step size for the case of direct quantization (dashed line) and the case of cascaded quantization (solid line). In the figure the variance of the source has been set to $\sigma^2 = 1.0$. In the case of cascaded quantization the step size is that of the second quantizer, Δ_2 . The step size of the first quantizer has

been kept fixed at $\Delta_1 = 1.0$. It is interesting to see that the cascaded quantizers may have a better coefficient-SNR than the direct quantization, for instance for the value $\Delta_2 = 0.756$. The TM5 quantizer is not designed to give the best coefficient-SNR at a certain step size. As will be seen further on, it does perform quite well in terms of entropy for a certain amount of distortion.

Fig. 15 shows the entropy as a function of the step size, using the same conventions as in Fig. 14. The effects of cascaded quantization are quite conspicuous in the figure. Observe that in the case of cascaded quantization certain values for the entropy are no longer attained. There will be gaps in the range of attainable entropies.

Fig. 16 shows the coefficient SNR as a function of the entropy for two different entropies in the quantization of the first quantizer. The two lower graphs show the extra error resulting from transcoding, that is, the error equals the difference

$$E = \text{SNR}_{\text{direct}} - \text{SNR}_{\text{transcoding}} \text{ (dB)}.$$

In the case of a relatively high input entropy H_1 there is a finer quantization in the first quantizer and, hence, more values are attainable for the entropy H_2 . This is confirmed by the figure. The figure also shows that, especially for relative low input entropies, some values for Δ_2 are less favorable than others. In the case of the almost vertical bar at approximately 1.4 bit/symbol in the left graphs, different coefficient SNR values may be obtained for the same entropy depending, on the choice of Δ_2 .

Fig. 17 shows the SNR as a function of the entropy for an I picture; in this figure the results for 64 coefficients are combined. In order to be able to compute this curve, the variances for all coefficients had to be known. These variances were measured in the test sequence ‘Mobi’. The step sizes in the first quantizer were as defined by the MPEG default intra weighting matrix. The DC coefficient has been excluded from the figure because in I pictures the step size for quantizing DC-coefficients is fixed and will, hence, be transcoded losslessly.

As in the case of individual coefficients, there are gaps in the range of attainable entropy values. Observe in Fig. 17 that the extra loss due to transcoding is greatest when the entropy H_2 is about

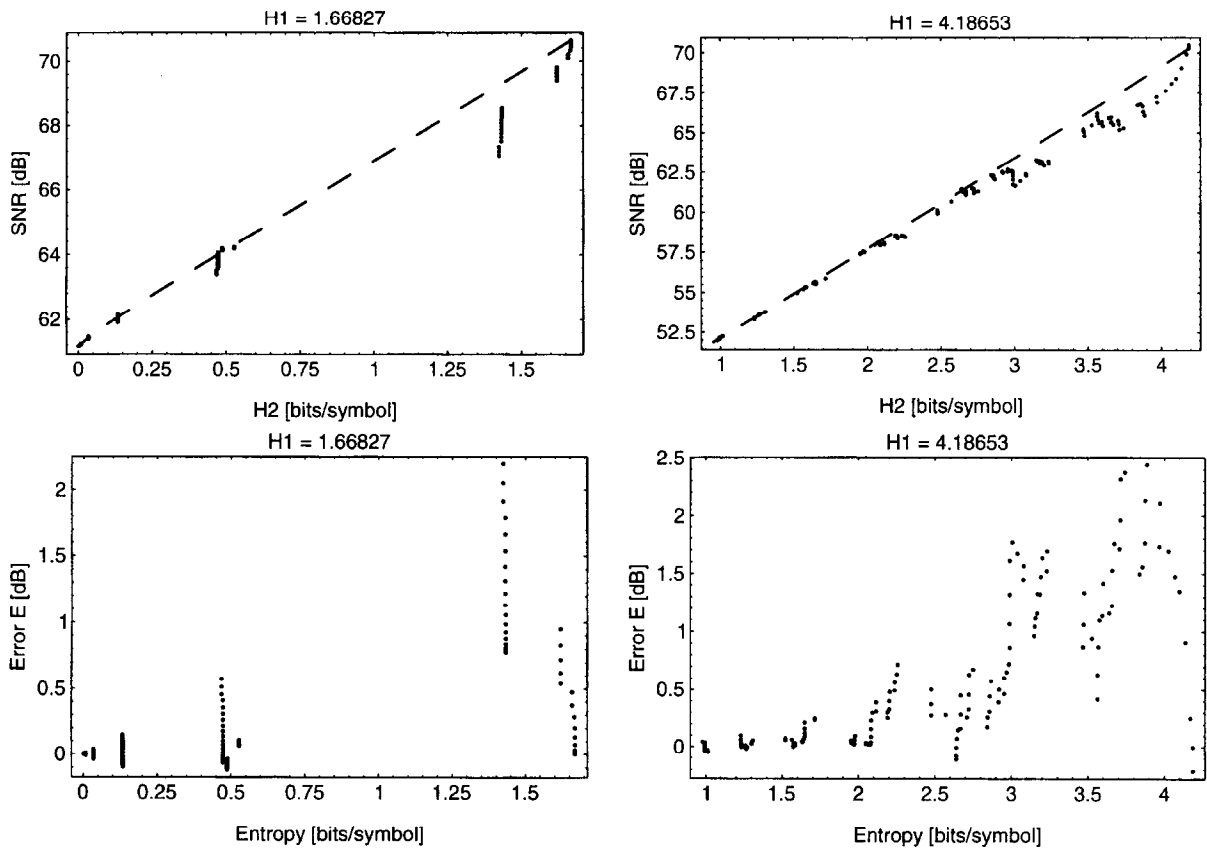


Fig. 16. (Top) The coefficient SNR as a function of the entropy. The graphs are obtained by means of computations. The dashed line represents direct quantization and the dots cascaded quantization. In all curves $\Delta_1 = 1.0$ holds. (Bottom) The difference between direct quantization and cascaded quantization, that is, the error E (dB). The two left graphs represent a 'low' entropy H_1 , whereas the two right graphs represent a 'high' entropy H_1 .

15% lower than H_1 , which corresponds to 166 bits/block. When the incoming bit-rate increases, the extra loss diminishes, as may be indeed expected because the influence of the first quantizer on the effective quantizer will diminish when its stepsize decreases.

3.4. Analysis of cascaded quantization with prediction loops

The results of the previous section are valid for I pictures. In the case of P pictures, however, the influence of the prediction loops has to be taken into account. In this section a quantization model

for cascaded quantization with prediction loops will be compared with direct quantization with prediction loops.

Fig. 18 shows a quantization model that incorporates the prediction loops to a certain level. In the model it is assumed that all motion vectors are zero, so that the memory and motion compensation can be simply modelled as a delay. This delay is indicated in the figure by the blocks ' z^{-1} '.

This model can be obtained from Fig. 9 in the following way. Since the motion compensation has been simplified to a simple delay the DCTs and IDCTs may be moved around in the block diagram of Fig. 9 in such a way that one DCT at the input and one IDCT at the output of the transmission

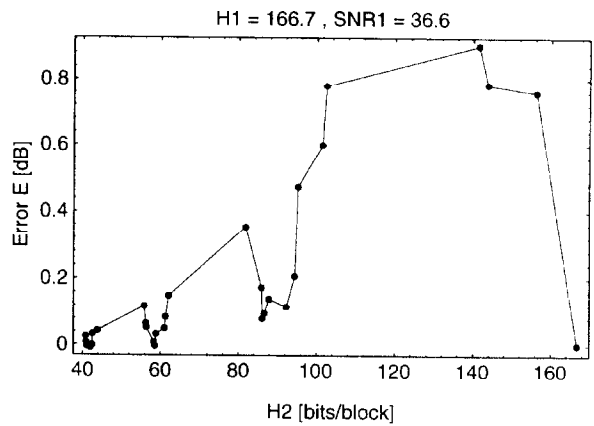
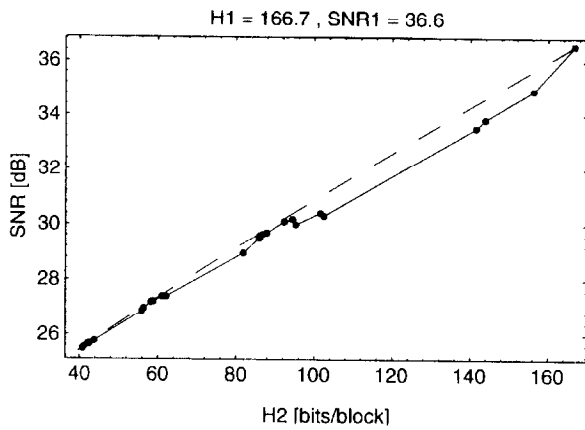


Fig. 17. The resulting SNR as a function of the entropy for an I picture and the error. The graphs are obtained by means of computations. Note: the lines refer to the order of the points with an increasing step size and are, hence, not function values. The rightmost dot has the lowest Δ_2 and Δ_2 increases from right to left.

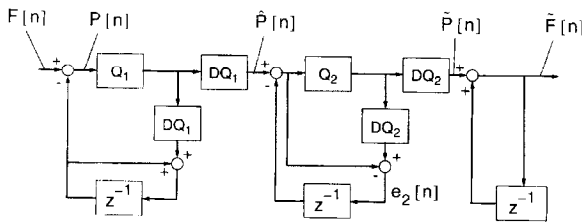


Fig. 18. The quantization model used for cascaded quantization in the case of P pictures.

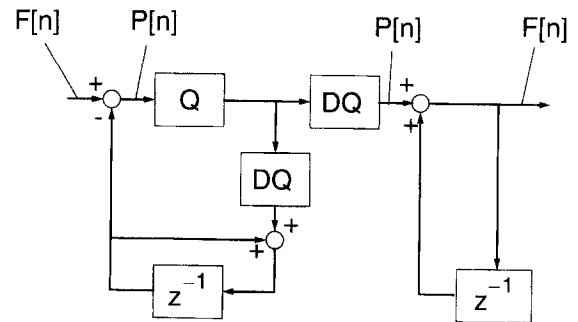


Fig. 19. The reference quantization model used for direct quantization in the case of P pictures.

chain will be present. In this case the quantization model of Fig. 18 represents the operation that is performed on each coefficient separately. Thus, this model enables us to compute the performance for each coefficient independently and add the results in the end.

Fig. 19 shows the quantization model for direct quantization. This will be the reference model. The performance of the cascaded quantization model of Fig. 18 can be compared with the performance of the direct quantization model of Fig. 19. Since the prediction loops make computations too complex the comparison is done using simulations.

These simulations were done as follows. It was assumed in these simulations that the pdf of the DCT coefficients in the residual picture have a two-sided Laplace pdf. The variances of each coefficient used by the random generator were ob-

tained from a measurement using the test picture ‘Mobi’ applying motion prediction and the DCT. By means of a random generator, a series of residual coefficients were produced. Using these artificially generated residual signals the input sample values were computed. It was assumed that a sequence of picture types ‘I–P–P–P’ was coded.

The cascaded quantization model and the direct quantization model are both based on the same artificially generated input signal. For both models, a series of points (H , SNR) are produced in the following way. For the cascaded quantization model the step size of the first quantizer is set to a fixed value $\Delta_1 = 7$, which corresponds to the

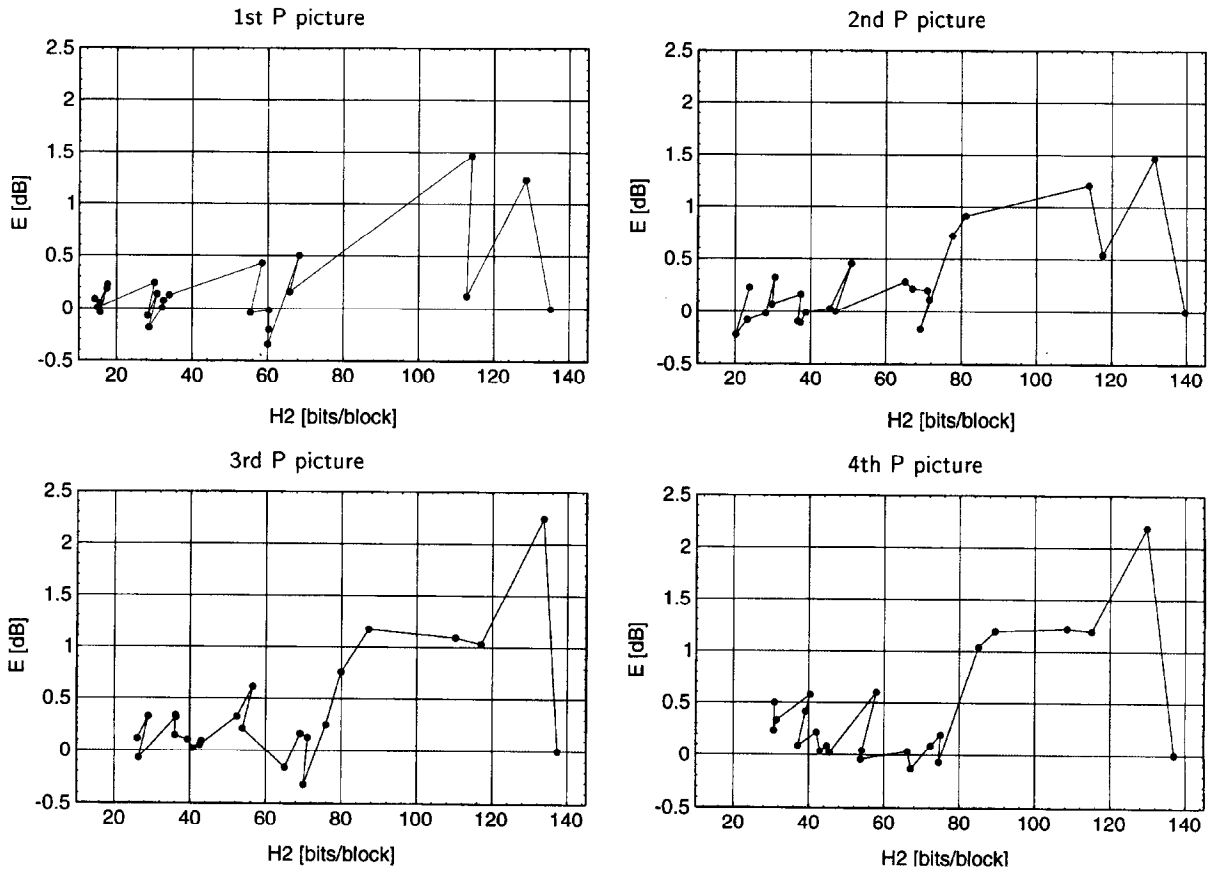


Fig. 20. The error for transcoding P pictures at several distances from the last I picture. The graphs are obtained by simulations with the models shown in Figs. 18 and 19. Note: the lines connect successive step sizes Δ_2 , not function values. The rightmost dot has the lowest Δ_2 and Δ_2 increases from right to left from [7, ..., 30].

average step size used for $R_1 = 15$ Mb/s. The step size of the second quantizer Δ_2 is varied in the range 7–30. For each of these step sizes Δ_2 a sufficient number of simulations is performed so that the entropy and distortion can be obtained. Hence, for each step size a point (H, SNR) is obtained.

A similar procedure is followed for the direct quantization model, i.e. the reference. Hence, a set of points (H, SNR) becomes available. It is assumed that the entropy-distortion function is a smooth function so that all the between values of the entropy can be calculated. By means of this calculation the performance can be compared at the same entropy values as obtained with the cascaded quantization model. For these entropy values the extra error E is computed from the difference in the SNR between the cascaded quantization model and

the direct quantization model. The results are shown in Fig. 20.

Fig. 20 shows that, as in the case of I pictures, the extra loss due to transcoding is most severe for an incoming bit-rate R_1 that is about 10–20% higher than the outgoing bit-rate R_2 . The figure also shows that there is a relatively small increase in loss in the case of P pictures at a larger distance from the last I picture. On the whole the results are quite similar to the results obtained for I pictures, shown in Fig. 17.

4. Measurements of the transcoder performance

The next step is to determine the performance of an actual transcoding system. This can be done

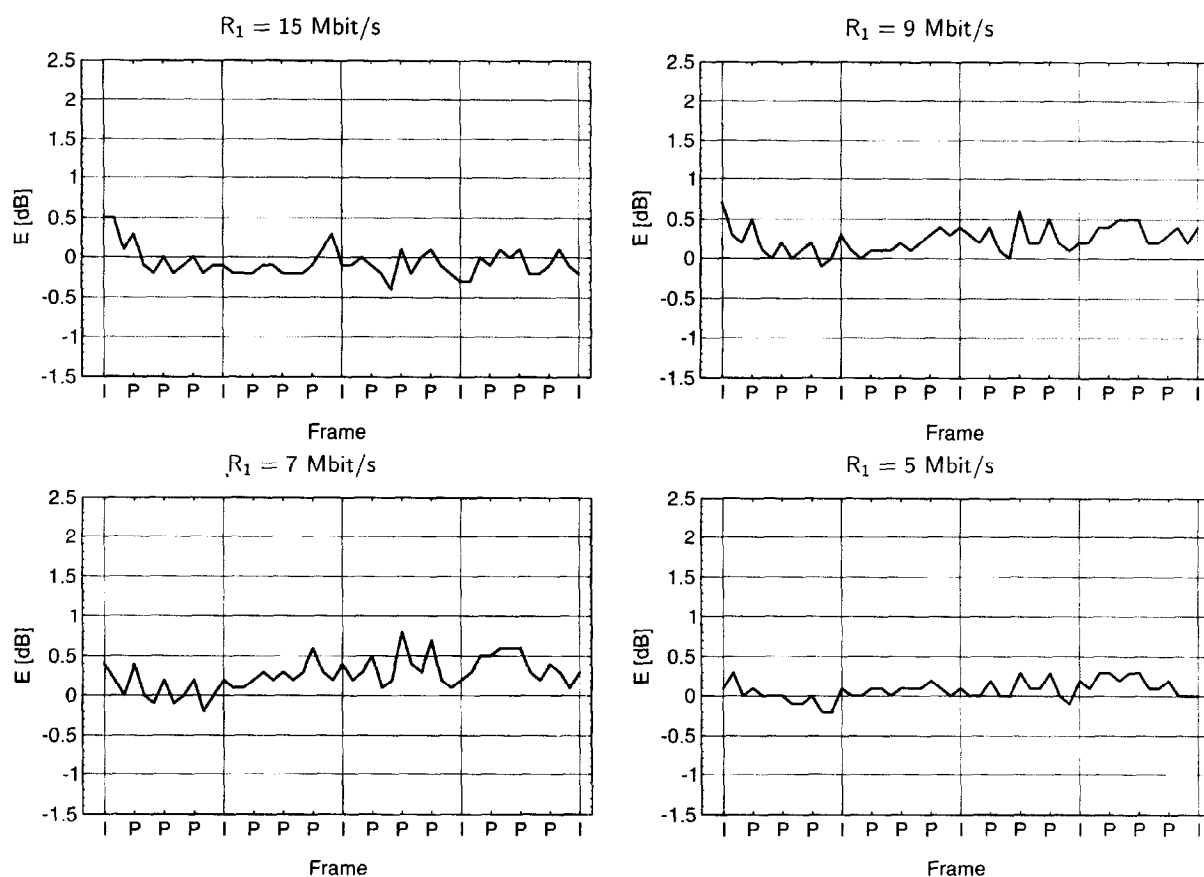


Fig. 21. Extra error in the case of transcoding at $R_1 = 15, 9, 7$ and 5 Mbit/s for 'Mobi' and $R_2 = 4.5$ Mbit/s. These results have been obtained by simulations with the complete transcoder.

with the aid of an experimental setup with computer simulations. The transcoder shown in Fig. 7 is implemented in software with the inclusion of many MPEG features. The features taken into account are: I, B and P pictures, field/frame prediction, field/frame coding, forward/backward/interpolated macro-block modes, coded/noncoded macro-block modes, skipped macro-blocks. The input and output bit-streams were syntactically correct MPEG-2 bit-streams. The settings of all these features are copied from the incoming bit-stream in order to minimize the transcoder complexity, that is, to facilitate the realization of Fig. 7 (see Section 2). The bit-rate control was similar to the one used in MPEG-2 TM5 [8]. In order to be able to make the comparison with stand-alone encoding, the glo-

bal targets were copied from a stand-alone encoding at the outgoing bit-rate R_2 .

The results of these transcoding simulations are shown in Fig. 21 for 'Mobi' and in Fig. 22 for 'Teeny'. The simulations were performed for a GOP structure 'I BB P BB P BB P BB', an input bit-rate of 5, 7, 9 and 15 Mbit/s and an output bit-rate of 4.5 Mbit/s. Note that the horizontal axes in the figures do not show B marks; however, the corresponding values for B pictures are also shown in the figures. The greatest loss results at an input bit-rate of approximately 7 Mbit/s. Note that this does not exactly correspond to the 10–20% mentioned above. The reason is that the earlier prediction was based on a simulation using the assumptions of a stationary Laplace-distributed input

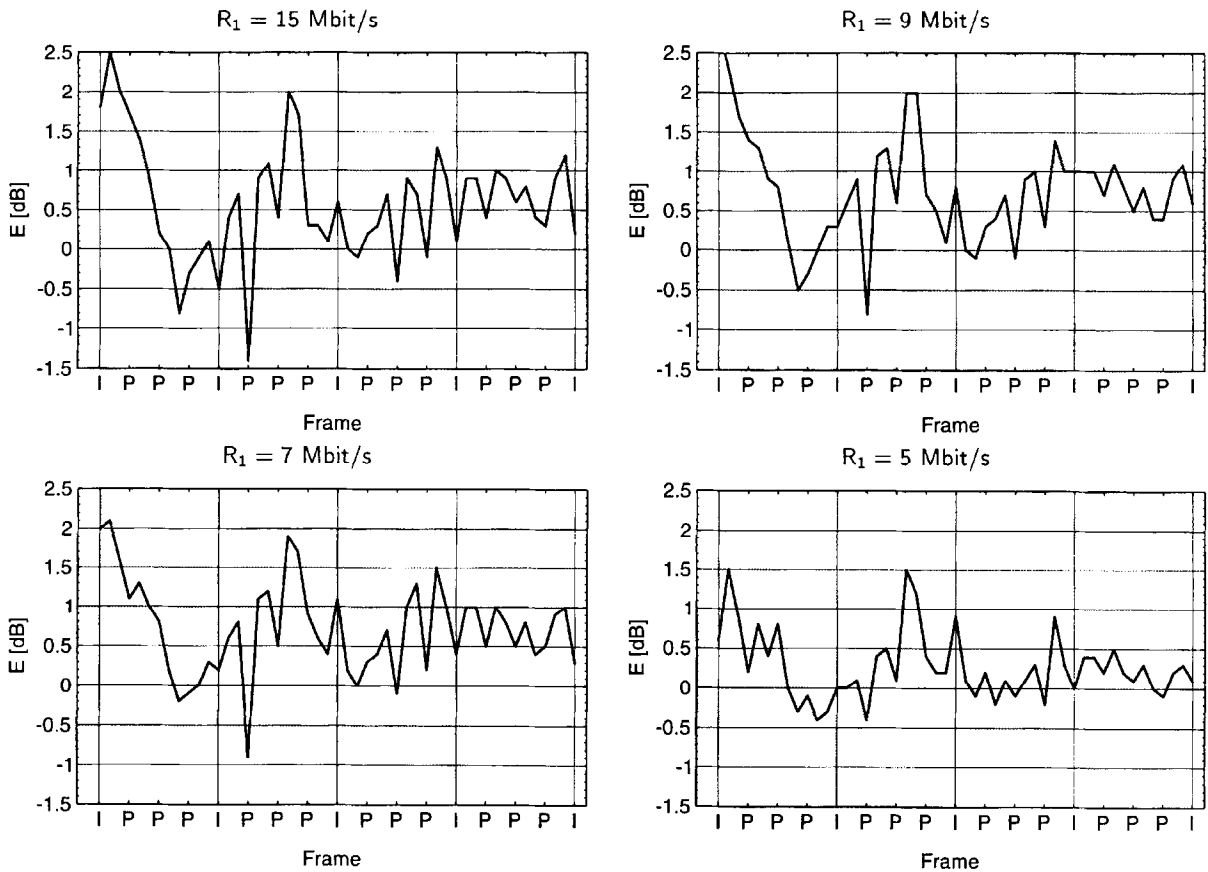


Fig. 22. Extra error in the case of transcoding at $R_1 = 15, 9, 7$ and 5 Mbit/s for ‘Teeny’ and $R_2 = 4.5$ Mbit/s. These results have been obtained by a simulation of the complete transcoder.

signal which may not have been completely valid in the actual simulation of the transcoder. Fig 23 shows similar data as Fig 21, except that it is a density plot. The horizontal axis shows the time and the vertical axis shows the input bit-rate to the transcoder. The output bit-rate was 4.5 Mbit/s in all cases. The dark shaded areas in this figure represent relatively large losses and the light shades represent small losses. The data in the figure are for 11 different bit-rates ranging from 5 to 15 Mbit/s. Fig 23 shows that the greatest losses occur at input bit-rates to the transcoder of 7 and 8 Mbit/s. The dark areas in the first pictures are due to transient effects of the transmission chain.

In all cases the loss is moderate for ‘Mobi’, that is, about 0.5 dB on average. For ‘Teeny’ the loss is

slightly larger, that is, about 0.8 dB on average. With ‘Teeny’ the variation in the quality loss is, however, greater, i.e. it ranges from -1 to $+2$ dB. The extra distortion due to transcoding is just noticeable in the case of the decoded pictures of ‘Mobi’. The appearance of the errors is similar as that resulting from ‘normal’, i.e. direct, coding. In the case of ‘Teeny’ the distortion was too small to be seen.

5. Conclusions

Transcoding may be necessary in some cases, for instance in the case of the remultiplexing of incoming compressed signals at a cable head end. Basi-

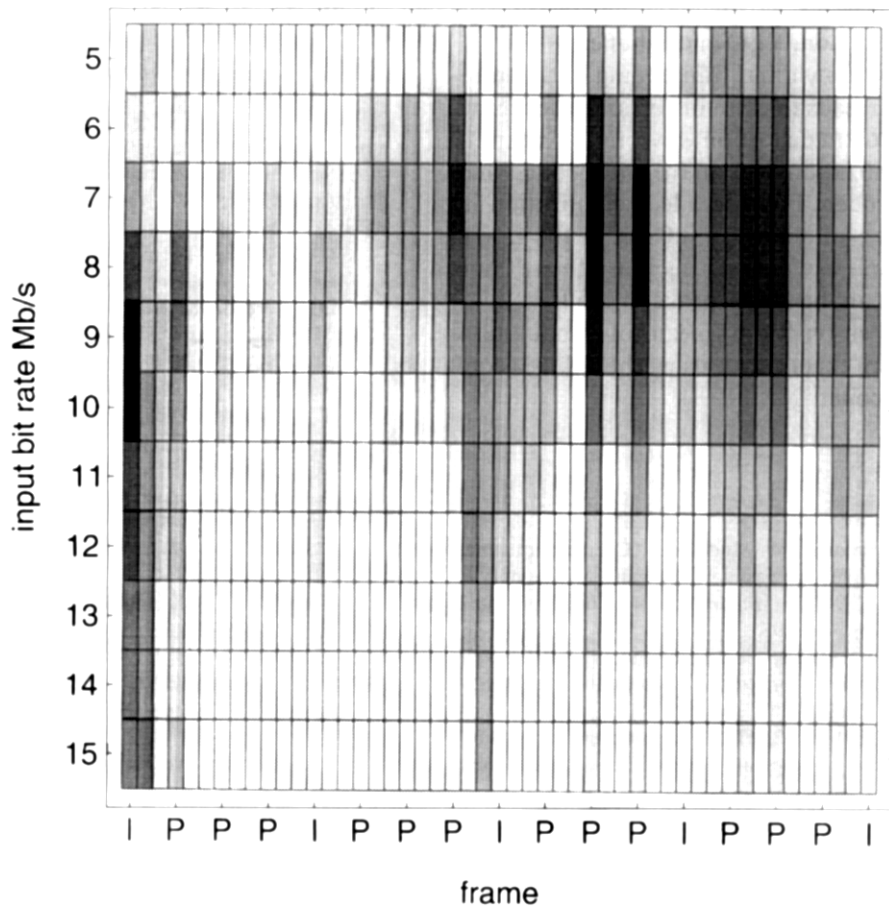


Fig. 23. Density plot of extra error with transcoding to 4.5 Mb/s for 'Mobi'. The darker areas refer to the greater losses.

cally, the transcoder used for such purposes consists of a cascaded decoder and encoder. This paper analyzes two problems involved in the use of a transcoder in the signal path. These problems are system complexity and system performance.

It has been shown in this paper that the complexity of the transcoder may be significantly less than that of a cascaded decoder plus encoder. It has been shown that several components in the cascaded decoder and encoder may be combined. In the proposed transcoder the incoming and outgoing picture types are equal, which means that motion vector and macro-block modes for the encoder can be copied from the incoming bit-stream.

A significant reduction in complexity is achieved by making use of the linearity of the motion compensation. This allows us to split the motion compensation in the encoder into two components, the motion compensation of the decoded picture and the motion compensation of the quantization error in the encoder. Because the motion compensation of the decoded picture is present in both the decoder and the encoder with opposite signs it is canceled in the transcoder. The conclusion is that one picture store for storing the prediction references instead of two is needed.

The transcoder derived in this paper has a complexity that is equivalent to the complexity of a de-

coder plus one DCT, quantization, de-quantization, a subtraction, VLC and a buffer.

It has been shown that transcoding in the transmission chain leads to extra distortion. The mechanism for this extra distortion was found to lie in the twofold quantization, once in the encoder and once in the transcoder. This cascaded quantization introduces losses of about 0.5–1.0 dB. In the case of a lower complexity pictures such as ‘Teeny’ there can occasionally be larger peaks in the loss.

This has been verified by means of computations and simulation experiments using the reduced complexity transcoder.

References

- [1] N. Farvardin and J.W. Modestino, “Optimum quantizer performance for a class of non-gaussian memoryless sources”, *IEEE Trans. Inform. Theory*, Vol. IT-30, No. 3, May 1984, pp. 485–497.
- [2] ISO-IEC/IS 11172-2, Coding of moving pictures and associated audio – for storage at up to about 1.5 Mbit/s, November 1992.
- [3] ISO/IEC IS 13818-2, ITU-T Recommendation H.262, Generic coding of moving pictures and associated audio information, November 1994.
- [4] N.S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall, London, 1984.
- [5] D. Le Gall, “MPEG: A video compression standard for multimedia applications”, *Commun. ACM*, Vol. 34, April 1991, pp. 46–58.
- [6] D. Le Gall, “The MPEG video compression algorithm”, *Signal Processing: Image Communication*, Vol. 4, No. 2, April 1992, pp. 129–140.
- [7] D.G. Morrison, M.E. Nilson and M. Ghanbari, “Reduction of the bit-rate of compressed video while in its coded form”, *6th Internat. Workshop on Packet Video*, Paper D17.1, Portland, Oregon, September 1994.
- [8] MPEG-2, Test Model 5 (TM5), Doc. ISO/IEC JTC1/SC29/WG11/93-400, Test Model Editing Committee, April 1993.
- [9] T. Nishitani, “Tandem transcoding without distortion accumulation”, *IEEE Trans. Comm.*, Vol. COMM-34, No. 3, March 1986, pp. 278–284.
- [10] A. Puri, “Video coding using the MPEG-2 compression standard”, *Proc. SPIE VCIP*, Boston, MA, November 1993, pp. 1701–1713.
- [11] R. Reiniger and J. Gibbs, “Distribution of the two dimensional DCT coefficients for images”, *IEEE Trans. Comm.*, Vol. COMM-31, No. 6, June 1983, pp. 835–839.
- [12] A.H. Wong and C.-T. Chen, “A comparison of ISO MPEG-1 and MPEG-2 video coding standards”, *Proc. SPIE VCIP*, Boston, MA, November 1993, pp. 1436–1448.