Image Communication, Special Issue on Real-time Video over the Internet, pp. 77-94, 15(1-2), September, 1999

# Robust Internet Video Transmission Based on Scalable Coding and Unequal Error Protection

Uwe Horn
Ericsson Eurolab, Herzogenrath, Germany

K. Stuhlmüller, M. Link, and B. Girod
Telecommunications Laboratory,
University Erlangen-Nuremberg,
Erlangen, Germany

**Abstract**

In this article we describe and investigate an Internet video streaming system based on a scalable video coder combined with unequal error protection that maintains an acceptable picture quality over a wide range of connection qualities. The proposed approach does not require any specific support from the network layer and is especially suited for Internet multicast applications where different users are perceiving different transmission conditions and no feedback channel can be employed. We derive a theoretical framework for the overall system by which the Internet packet loss behavior can be directly related to the picture quality perceived at the receiver. We demonstrate how this framework can be used to select appropriate parameter values for the overall system design. Experimental results show how that the presented system achieves a gracefully degrading picture quality for packet losses up to 30%.

## 1 Introduction

The demand for Internet video streaming services has rapidly grown over the past few years. Since the Internet supports real-time services only in a best effort manner, it remains to be a challenging task to design high-quality video streaming systems which are able to cope with the Internet's unpredictable and varying network conditions. Network conditions are becoming even more varying in the future if mobile packet networks like UMTS (Universal Mobile Telecommunications System, [1, 2]) are used for wireless Internet access.

The degradation in quality of an Internet video connection is mainly determined by the packet loss behavior observed at the video decoder. Congested routers are one typical source for packet losses. By posing a real-time constraint on a transmission system as meant by the term *streaming*, packets arriving too late at the receiver are another source of packet loss. The real-time constraint requires that a packet containing real-time data sampled at a time

instance $t_s$ needs to be presented to the user at the receiving site no later than at time instance $t_s + d$, where $d$ is the maximum acceptable overall end-to-end delay. The end-to-end delay $d$ includes all individual delays which are caused by the encoder, the network transmission, and the decoding process. Acceptable values for $d$ heavily depend on the type of application. For instance telephony applications require an overall round-trip delay not higher than 250 ms, whereas on-demand or multicast applications tolerate end-to-end delays up to several seconds.

Internet video streaming applications usually employ a variable bitrate coder. Since an Internet connection can be considered as a rate-limited transmission channel with varying delay, buffers at the sender and the receiver are required. Constraints on the encoder and decoder buffer sizes and the delay introduced by these buffers can be found in [29]. The derivations presented in [29] show how the overall end-to-end delay can be traded against an increased maximum allowed transmission delay. Especially in Internet streaming applications an additional packet loss is introduced by limiting the overall end-to-end delay which makes it difficult to realize delay critical applications like video telephony.

The Internet packet loss behavior and the increased amount of lost packets resulting from a constrained end-to-end delay can be illustrated by a transmission experiment in which we transmitted packets over the Internet between Erlangen, Germany, and Stanford, California, by using the Real-time Transport Protocol (RTP [3]). Over a period of 100 seconds we sent 200 packets per second, each packet 625 bytes long, resulting in an overall transmission rate of 1 Mbps. Fig. 1 shows the measured delay and loss for each packet and the overall histogram. We measured an overall packet loss percentage of 8.29%. From the top graph it can be seen that packet losses often occurred in bursts. The bottom graph shows the packet delay variation which was between 130 ms and 360 ms. If we limit the overall network transmission delay to 250 ms, the receiver perceives a slightly increased packet loss of 9.97% since packets delayed by more than 250 ms have to be discarded in this case.

To cope with the sometimes rapidly varying transmission conditions over Internet connections we propose and investigate in this article a video transmission scheme which combines a scalable video coder with unequal error protection across packets. Our approach does not require any support from the network and it can be employed in any packet oriented network, not only the Internet. Although the proposed approach may slightly increase the overall end-to-end delay, it is perfectly suited to improve the picture quality in video-on-demand and multicast applications where the overall end-to-end delay is less critical than for applications like video telephony.

Scalable video coders especially for Internet applications have been proposed in the past by many authors [4, 5, 6, 7, 8, 9, 10, 11]. The idea of combining scalable coding with unequal error protection has already been proven to result in gracefully degrading transmission schemes [12]. Applying forward error correction across video data packets has previously been proposed in the PET system [13, 14] for MPEG-1 Internet video transmission and also for lost cell recovery in ATM based video transmission [15, 16, 17]. A good overview about existing video streaming applications in the Internet is given in [18].

In this paper we present and investigate an Internet video streaming system which combines a truly scalable codec with unequal error protection across packets (Fig. 2). Our forward error correction scheme uses a fast and flexible Reed–Solomon codec implementation based on Newton's interpolation. Since the overall system behavior is characterized by many different parameters, we also describe a theoretical framework based on an analytically tractable

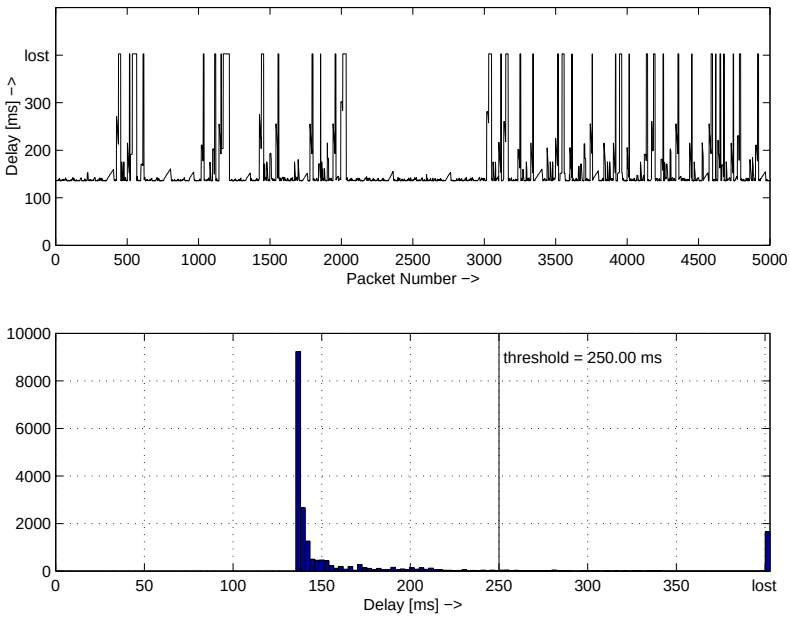


Figure 1: Measured delay and loss between the Telecommunications Laboratory in Erlangen, Germany, and Stanford University, California. Over a period of 100 seconds 200 packets per second each 625 bytes long were sent which results in a total channel data rate of 1 Mbps. *Top:* Delay vs. packet number for the first 5000 packets. *Bottom:* Delay histogram.

Internet packet loss model which allows us to relate all important system parameters to the overall picture quality at the video receiver. This framework is especially useful to analytically investigate and optimize the overall system behavior.

We have structured this article as follows. In section 2 we review an analytically tractable Internet packet loss model which we will use throughout the rest of this paper. Section 3 reviews Reed-Solomon codes and describes a fast software-only implementation of a Reed-Solomon channel codec. In section 4 the proposed transmission scheme including the scalable video codec employed is presented. We then describe in section 5 an analytical framework which relates all important system parameters to the overall picture quality perceived at the receiver. This framework is used to select the system parameters for the simulations finally presented in section 6.

# 2 Channel Model for Internet Video Streaming

The underlying processes which lead to packet losses over Internet connections are quite complex. However, even a simple and analytically tractable Markov model with only two states approximates the Internet's packet loss behavior fairly well. It has been used previously to model congestion situations in ATM networks [17]. The two states of the model are denoted G (good) and B (bad). In state G packets are received correctly and timely whereas in state B packets are lost. The model is fully described by the transition probabilities $p_{\mathrm{GB}}$ between states G and B and $p_{\mathrm{BG}}$ between states B and G. Since these parameters are not very intuitive,

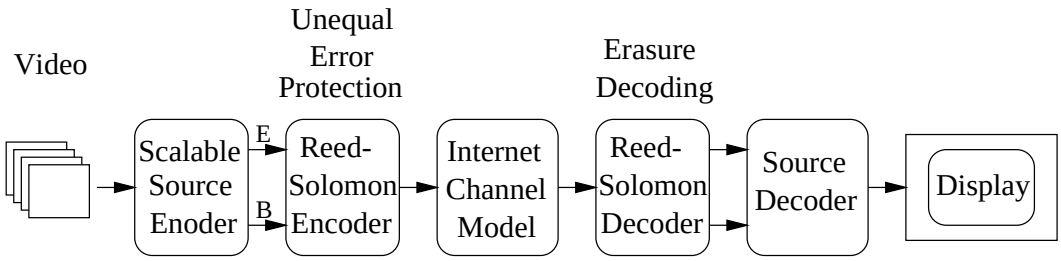


Figure 2: Outline of the video transmission scheme proposed and investigated in this article. A scalable source coder encodes the input video in two layers, a base layer B carrying the most important and an enhancement layer E carrying the less important video information. Before the layers are transmitted they are packetized and protected against packet loss according to their importance. Reed–Solomon codes are employed for forward error correction. A simple two-state packet loss model is used to simulate the typical packet loss behavior over Internet connections. At the receiver a Reed–Solomon decoder reconstructs for each layer as many original packets as possible which are finally fed into the source decoder where they are decoded and displayed.

we prefer to use the average loss probability

$$P_{\mathrm{B}} = \Pr(\mathrm{B}) = \frac{p_{\mathrm{GB}}}{p_{\mathrm{GB}} + p_{\mathrm{BG}}}$$

and the average burst length

$$L_{\mathrm{B}} = \frac{1}{p_{\mathrm{BG}}} \,,$$

which is the average number of consecutively lost packets. To get an idea of typical parameter values we calculate $P_{\mathrm{B}}$ and $L_{\mathrm{B}}$ for the transmission experiment in Fig. 1. Under the assumption of a maximum transmission delay of 250 ms we obtain $P_{\mathrm{B}} = 0.0997$ and $L_{\mathrm{B}} = 9.57$.

In other transmission experiments where we investigated various packet sizes, data rates and maximum transmission delays, we obtained for $P_{\mathrm{B}}$ values between 0 and 0.6 and for $L_{\mathrm{B}}$ values between 2 and 20. Low values of $P_{\mathrm{B}}$ occurred along with high values of $L_{\mathrm{B}}$ and vice versa. For a constant channel data rate an exchange between packet rate and packet size was possible as long as a packet rate of roughly 400 packets/s was not exceeded. Higher packet rates resulted in substantially increased packet losses.

# 3 Packet Loss Protection Based on Reed–Solomon Codes

The idea of forward error correction (FEC) in a packet oriented video transmission scheme is to generate redundant packets at the sender which can be used at the receiver to recover lost video data packets. The FEC scheme proposed in this article uses Reed–Solomon (RS) codes. These block codes are perfectly suited for error protection against bursty packet losses because they are maximum distance separable codes, i.e. there are no other codes that can reconstruct erased symbols from a smaller number of received code symbols [19].

An RS code can be easily employed for protecting packets against loss if the codewords are formed across $k$ information and $n - k$ redundancy packets as shown in Fig. 3. The resulting

$n$ packets are called a block of packets (BOP). While the packets are the rows of the BOP, the codewords are calculated across the columns. Note that the scheme depicted in Fig. 3 introduces no additional delay at the sender. The information packets can be sent as soon as they have been generated. The sender only has to store copies of the information packets until $k$ packets have been sent. Then the $n-k$ redundancy packets are calculated and transmitted just after the last information packet. The information packets can be reconstructed from any

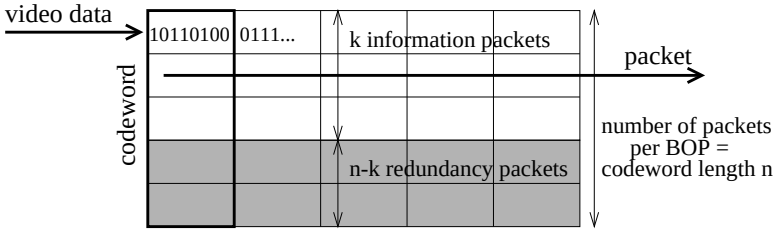


Figure 3: Example of a Block of Packets (BOP) with $k = 3$, $n = 5$.

subset of at least $k$ correctly received packets using erasure decoding [20]. This is possible because each packet contains its packet number, and therefore the exact positions of lost packets in the BOP are known at the receiver. As soon as any $k$ packets of a BOP have been received, all lost information packets can be reconstructed. Thus, our FEC scheme requires a receiver buffer which can at least hold $k$ packets.

In the following we present a simple definition of RS codes which allows us to interpret the RS coding and decoding process as an interpolation problem. For this problem, we then describe Newton's Interpolation as a flexible and fast solution which is especially well suited for software implementation.

## 3.1 Definition of Reed–Solomon Codes

An $(N, k)$ RS code over the Galois Field GF($2^m$) with codeword length $N$ and dimension $k$ is defined by the set of all polynomials $C(x)$ over GF($2^m$) with degree less than $k$ [19]. $k$ is the number of information symbols per codeword and $m$ is the number of bits per symbol. In the following we use $m = 8$, i.e. symbols are bytes. The codewords $[\tilde{c}_0, \tilde{c}_1, \ldots, \tilde{c}_{N-1}] \in \mathrm{GF}(2^m)^N$ result from sampling the polynomials $C(x)$ at the pair-wise distinct locations $\tilde{z}_i \in \mathrm{GF}(2^m), 0 \leq i \leq N-1$, i.e. $\tilde{c}_i = C(\tilde{z}_i)$. $N$ must divide $2^m - 1$, but it is very easy to get an $(n, k)$ RS code with codeword length $n \leq N$ by simply dropping redundancy samples, a method commonly referred to as puncturing [19]. Thus, the codeword length is only restricted by the condition $n \leq 2^m - 1$ and, since we are dealing with bytes, $n \leq 255$. The code rate of the $(n, k)$ RS code is $k/n$, which is the fraction to which the source data rate has to be reduced in order to maintain a constant channel data rate.

## 3.2 Newton's Interpolation for Reed–Solomon Channel Coding and Decoding

In the previous subsection we defined an $(N, k)$ RS code by the set of all polynomials $C(x)$ with degree less than $k$. The calculation of redundancy symbols at the encoder and the reconstruction of lost information symbols at the decoder can be treated as basically the same

*interpolation* problem: compute unknown samples of $C(x)$ at the locations $z_i, k \leq i \leq N-1$, using the $k$ given samples $c_i = C(z_i), 0 \leq i \leq k-1$. In general the locations $\{z_i\}$ are a re-sorted subset of the original locations $\{\tilde{z}_i\}$. In our application the locations are simply the packet numbers within the BOP, and the samples are the data bytes in the packets.

Any interpolation algorithm could be used. We have chosen *Newton's interpolation* [21], since it is a fast and flexible algorithm with virtually free choice of $k$ and $n \leq N$. In the following we briefly review Newton's interpolation because of its relevance for the implementation of our RS channel codec.

The basic idea of Newton's interpolation is to use an iterative approach where the degree of the interpolation polynomial is increased by one in each iteration:

$$\begin{aligned}
&\text{Degree } 0: && C_0 \\
&\text{Degree } 1: && C_0 + C_1(x - z_0) \\
&\text{Degree } 2: && C_0 + C_1(x - z_0) + C_2(x - z_0)(x - z_1) \\
&\quad \ldots \\
&\text{Degree } N-1: && C_0 + \sum_{j=1}^{N-1} C_j \prod_{i=0}^{j-1} (x - z_i) \qquad (1)
\end{aligned}$$

The $C_j, 0 \leq j \leq N-1$, are called the *Newton coefficients* of $C(x)$. As $C(x)$ is a polynomial of degree less than $k$, the coefficients $C_j$ are equal to 0 for $j \geq k$. Note that the Newton coefficients depend on the particular set and order of the locations $z_i$, while the degree $k-1$ interpolation polynomial is exactly the original polynomial $C(x)$. Thus, any subset of $k$ samples can be used in any (but known) order.

The Newton coefficients can be calculated using the matrix of divided differences $C_{i,j}$ [21] shown in Fig. 4. The $k$ known samples are fed into the first line of the matrix, which is then calculated from top to bottom. The dependencies are marked by arrows. The Newton coefficients appear in the main diagonal of the matrix.

Note that a sample $c_m$ influences the coefficients $C_{i,j}$ with $i \geq m$ only. This is the reason for the flexibility of the scheme: on the left of the dotted line in Fig. 4, with every arriving symbol $c_m$ from location $z_m$, the next column $C_{m,j}, 0 \leq j \leq m-1$, can be computed. Neither it is necessary to know the following symbols $c_i$ nor their locations $z_i, i \geq m+1$.

As soon as $k$ symbols have been processed, we can compute any of the $N-k$ unknown samples $c_i$ at the locations $z_i, i \geq k$. This is shown on the right of the dotted line in Fig. 4: As the Newton coefficients $C_j$ are all zero for $j \geq k$, the coefficients $C_{i,j}$ are zero for $i \geq k$, and the rest of the matrix is calculated from bottom to top. The interpolated samples appear in the first row.

The complexity of Newton's Interpolation is determined by how many $C_{i,j}$ have to be calculated (Fig. 4). When $u$ denotes the number of unknown samples that are calculated, then the number of coefficients $C_{i,j}$ to be calculated is

$$\frac{k(k-1)}{2} + u(k-1) = (k-1)(\frac{k}{2} + u),$$

where $u = n - k$ at the encoder and $u$ is the number of lost video data packets at the decoder.

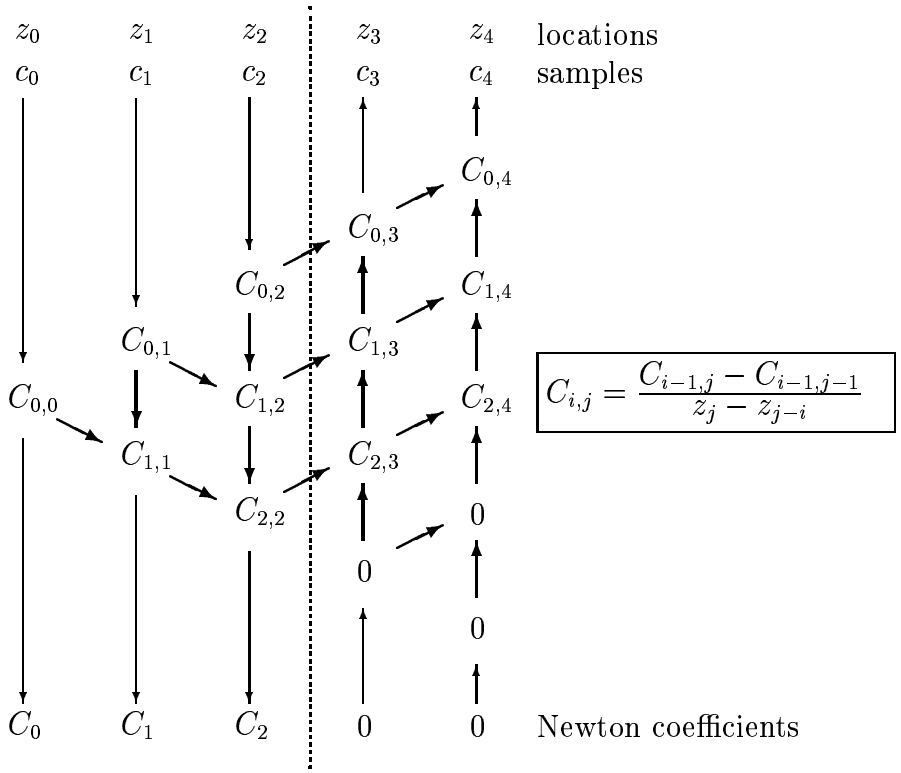


Figure 4: Matrix of divided differences for an $(n = 5, k = 3)$ RS code. *Encoder:* On the left of the dotted line, the Newton coefficients $C_0, \ldots, C_{k-1}$ are calculated from the information symbols $c_0, \ldots, c_{k-1}$. The dependencies are given by the formula in the box and marked with arrows. As, by definition, an $(n, k)$ RS polynomial is of degree less than $k$, the Newton coefficients $C_j$ are set to zero for $j \geq k$. Then, on the right of the dotted line, additional samples $c_k, \ldots, c_{n-1}$ can be calculated. These are the redundancy symbols. *Decoder:* Any $k$ received symbols $c_0, \ldots, c_{k-1}$ are used to calculate the Newton coefficients $C_0, \ldots, C_{k-1}$. Note that in case of lost information packets these symbols and coefficients are not identical with the ones at the encoder. The locations $z_0, \ldots, z_{k-1}$ are now given by the packet numbers within the BOP. The lost information symbols are calculated as samples at the missing locations.

# 4 Scalable Video Coding Combined with Unequal Error Protection

In this section we describe our proposed Internet video transmission scheme which employs a two-layer scalable video coder. The base layer is encoded with a fully standard compatible H.263 coder whereas the enhancement layer is encoded by an $E_8$-lattice quantizer explained later in this section. For Internet packet video transmission both layers are protected by forward error correction based on Reed-Solomon codes introduced in section 3. We explain in detail the packetization of the layered video stream and how Reed-Solomon channel coding across packets is applied for unequal error protection (UEP).

## 4.1 Scalable Video Codec

The scalable video codec employed in our transmission scheme is based on a spatio-temporal resolution pyramid first proposed in 1991 by Uz and Vetterli [22]. It consists of two spatial resolution layers corresponding to QCIF ($176 \times 144$ pels) and CIF ($352 \times 288$ pels) resolution (Fig. 5). The QCIF base layer is encoded at a temporal resolution of 15 frames per second (fps) whereas the CIF layer is encoded at 30 fps. Within each layer an Intra-frame is encoded every 266 ms.

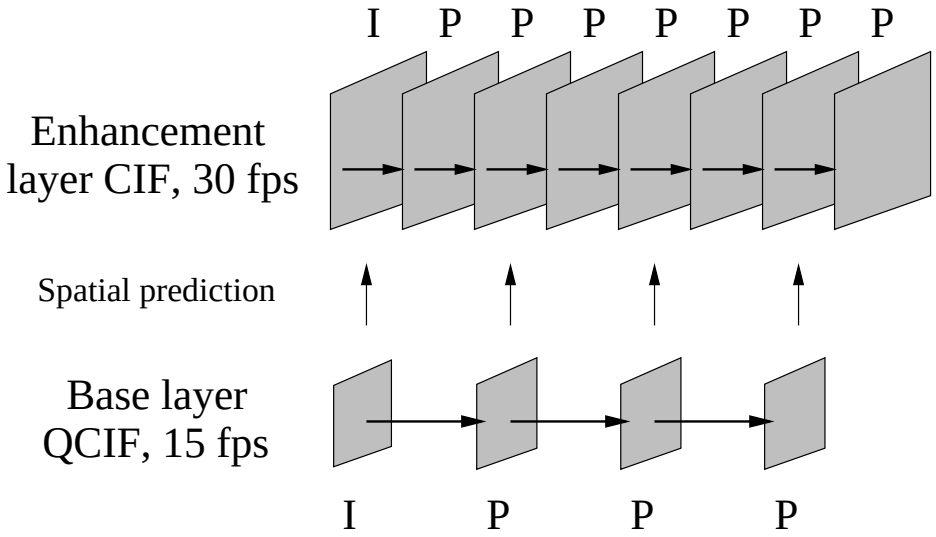


Figure 5: The group of picture (GOP) structure of the scalable coder used throughout this article. It consists of two layers, a base layer at QCIF resolution complemented by an enhancement layer at CIF resolution. Intra-pictures are denoted by I, predictively coded Inter-pictures are denoted by P as usual. This GOP structure repeats every eight frames, meaning that in each layer an Intra-frame is transmitted every 266 ms. Spatial and temporal redundancies are exploited by applying spatial and/or temporal prediction as denoted by vertical and horizontal arrows.

The hierarchical coder used to encode the spatio-temporal resolution pyramid of Fig. 5 is shown in Fig. 6. Each input frame is first downsampled to QCIF resolution by a simple (1/2 1/2) filter separately applied in horizontal and vertical direction. The QCIF base layer is encoded by a fully standard compatible H.263 encoder. The reconstructed QCIF image at the encoder is interpolated by an (1/4 3/4 3/4 1/4) filter which again is separately applied in the horizontal and vertical direction. The interpolated QCIF image is then fed into an enhancement layer coder (EL-Coder) which is shown in detail in Fig. 7.

In Intra-mode the EL-Coder encodes the spatial prediction error between the interpolated, reconstructed QCIF image and the original input image. In Inter-mode the temporal prediction error after motion compensation within the CIF layer is encoded. As in most of the standardized codecs, motion compensation is carried out with half-pel accuracy based on $16 \times 16$ blocks.

The decision between Intra- and Inter-frame mode for each macroblock in the P-frames is based on a Rate-Distortion (RD) criterion in the lattice as well as in the H.263 DCT based coder. For each macroblock the coding mode with the smallest Lagrangian cost $D + \lambda R$

is selected. $D$ denotes the mean squared error (MSE) and $R$ denotes the bit rate resulting from the selection of the corresponding coding mode. Simulations have shown that letting $\lambda = 0.85 * q^2$, with $q$ denoting the quantizer step size parameter, outperforms simpler mode selection schemes which disregard the rate by assuming $\lambda = 0$ and which often use the sum of absolute differences (SAD) instead of the MSE as distortion measure [23].

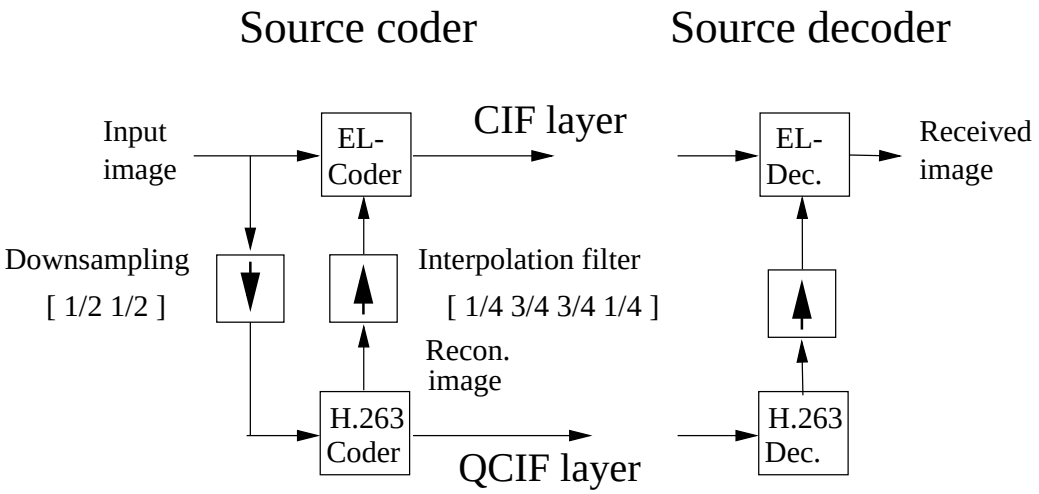


Figure 6: Hierarchical codec for encoding the spatio-temporal resolution pyramid of Fig. 5. The input image is first downsampled to QCIF resolution. The base layer is then encoded by a fully standard compatible H.263 coder. The locally reconstructed H.263 image is interpolated and fed into the enhancement layer encoder (EL-Coder) which Fig. 7 shows in more detail.

The motion vectors required for the CIF layer are estimated hierarchically based on those found by the H.263 coder in the QCIF layer. In the QCIF layer the motion vector search window is $\pm 15$ pels. In the CIF layer only refinement motion vectors found within a $\pm 3$ pels search window are transmitted [24].

Depending on the selected coding mode either a spatial or a temporal prediction error signal is quantized and encoded by the $E_8$-lattice quantizer ($E_8$-LTQ) shown in Fig. 8. We selected the $E_8$ lattice since it corresponds to the densest sphere packing in eight dimensions [25]. Furthermore, there exist fast algorithms for nearest lattice point rounding which are described, e.g., in [26, 27].

For encoding, a $2 \times 4$ block of neighboring samples is mapped into an 8-D vector. This vector is then scaled by a factor $1/s, s >> 1$ where $s$ corresponds to the quantizer step size in one-dimensional quantization and is used to control the bit rate of the enhancement layer. The scaled 8-D vector is rounded to its nearest $E_8$-lattice point by using the algorithm given in [26]. From the lattice point an index is computed which then is transmitted to the decoder. The decoder reconstructs the lattice point from the received index either by computation or by a simple table-lookup within a codebook for the most probable vectors. This codebook is used to adapt the lattice quantizer to the statistics of the spatial and temporal prediction error signals which results in an overall increased coding efficiency. Finally, by rescaling the reconstructed lattice point with $s$, the reconstructed $2 \times 4$ input block is obtained.

Note that the scalable video codec is very similar to the spatial scalability mode proposed in Annex O of H.263+ [28]. The main difference is the use of an $E_8$-lattice quantizer instead

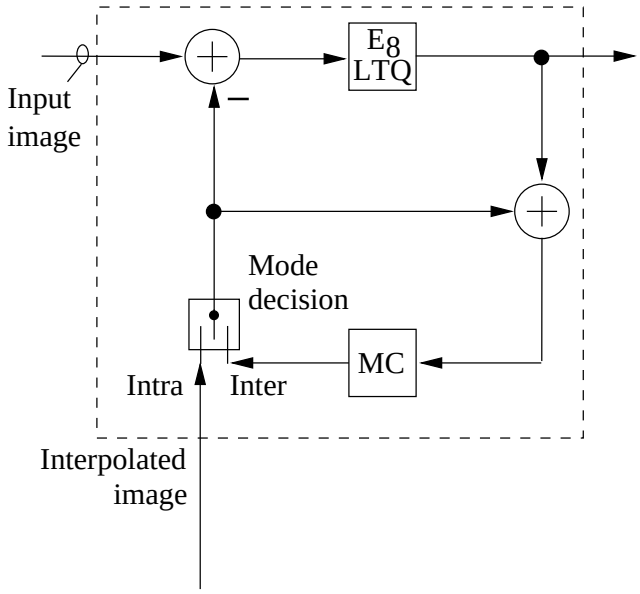


Figure 7: Enhancement layer encoder (EL-Coder) employed in the hierarchical codec of Fig. 6. In Intra-frame mode the spatial prediction error is encoded. In Inter-frame mode the prediction error after motion compensation is encoded. Mode decision is carried out for each macroblock. Instead of an $8 \times 8$ DCT used in standardized codecs, an $E_8$-lattice quantizer (Fig. 8) is used for encoding the prediction error signals.

of the $8 \times 8$ DCT to encode the enhancement layer signal. The interpolation filter is exactly the same as proposed in H.263+. Simulation results presented in [11] show that combining an efficient DCT based base layer encoder with the proposed $E_8$-lattice coder for the enhancement layer results in a higher coding efficiency and a faster decoder than the spatial scalability mode of H.263+. Nevertheless the main results obtained in section 6 are applicable as well to a H.263+ compliant scalable codec.

## 4.2 Packetization and Unequal Error Protection Across Packets

This subsection describes how the hierarchically encoded video stream is packetized, protected by unequal error protection and transmitted to the receiver. Note that in general we are not limited to two layers. The proposed packetization scheme can be extended easily to support any required number of layers or any type of data partitioning that benefits from unequal error protection.

At the encoder we multiplex the bitstreams of all layers into one block of packets (BOP) as shown in Fig. 9. We assume $n$, the number of packets in one BOP, to remain fixed. However, to cope with variable bit rate transmissions we allow varying packet sizes in different BOPs. The bits belonging to layer $l$ are filled into $k_l$ packets. The remaining $(n - k_l)$ packets in the BOP are filled with channel coding redundancy as described in section 3. Therefore, $k_l$ determines the protection level of layer $l$.

It is advantageous to pack each GOP (group of pictures, see Fig. 5) into a fixed number of BOPs (or vice versa) in order to be able to place synchronization words at the beginning of the video data fields of each layer in each BOP. In the experiments presented in this paper

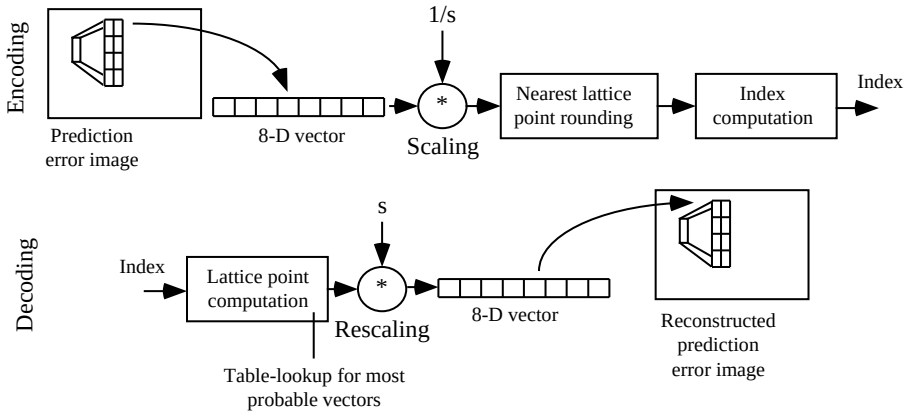


Figure 8: $E_8$-lattice codec used within the enhancement layer encoder of Fig. 7. Top: Obtaining an index for a $2 \times 4$ block of neighboring samples of the prediction error image. Bottom: Reconstruction of a $2 \times 4$ block from the received index. Coding efficiency can be increased by using a codebook for the most probable vectors.

it was always possible to pack exactly one GOP into one BOP.

The structure of each BOP, e.g. how many layers are transmitted and how video data and channel coding redundancy of each layer are distributed within the BOP, needs to be transmitted as side information to the decoder. We use an *Info* field at the beginning of each BOP which contains:

- the number $L$ of layers in this BOP (1 byte),
- for each layer $l$, $0 \leq l \leq L-1$:
  - the number $k_l$ of packets that contain video data for layer $l$ (1 byte),
  - the number $S_l$ of bytes that layer $l$ occupies in each packet (2 bytes).

This side information can be transmitted such that it is protected at least as strongly as the base layer. The BOP packets themselves are transmitted as RTP packets. For recovering the information at the decoder we need to know the BOP number, which can be transmitted by adding 1 byte in the RTP payload header. The position of the packet in the BOP can then be derived from the RTP sequence number.

The overall channel data $R$ rate of the scalable video coding scheme with UEP is

$$R = \sum_{l=0}^{L-1} \frac{n}{k_l} \cdot V_l \,, \tag{2}$$

where $V_l$ is the average video data rate of layer $l$ and the fixed contribution of the small *Info* block is ignored.

The proposed UEP packetization scheme introduces an additional delay, which will be discussed briefly in the following. A detailed discussion of delay-related topics in video streaming applications can be found in [29].

At the sender, we need a buffer between video encoder and RS encoder (see Fig. 2) which can hold the video data of one BOP, because we optimize the packet size parameters $S_l$ (see

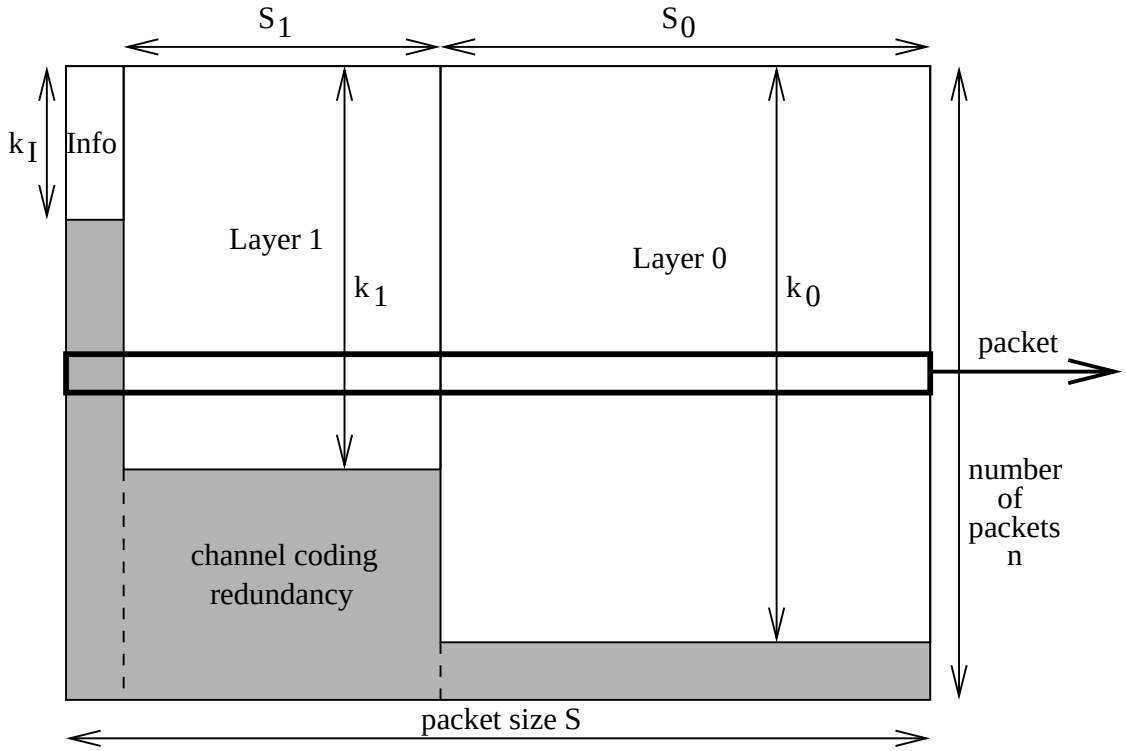


Figure 9: Block of packets (BOP) layout for transmitting a layered video stream consisting of $L = 2$ layers. The video stream belonging to layer $l$ is distributed across $k_l$ packets and occupies $S_l$ bytes in each packet. The shaded areas are filled with channel coding redundancy.

Fig. 5) for each BOP such that the BOP contains as less unused bytes as possible. This optimization is required since we assume variable bit rate video coding. The buffer introduces a delay of $1/R_{\mathrm{BOP}}$, where $R_{\mathrm{BOP}}$ denotes the BOP rate which is the number of BOPs processed per second. In applications where a low delay is more important than bitrate efficiency, this optimization step could be omitted by fixing maximum values for $S_l$ in advance, thereby accepting video data fields which are not completely filled in some BOPs. In this case no additional delay is introduced at the sender.

At the receiver, a buffer is required before the RS decoder because erasure decoding as described in section 3.2 is possible only if at least $k$ packets of a BOP were received. Therefore, the decoder buffer must be able to store at least $k$ packets and introduces a delay of $1/R_{\mathrm{BOP}}$. Note that this decoder buffer can be part of the receiver buffer, which is required anyway as discussed in section 1.

In our implementation $R_{\mathrm{BOP}}$ is fixed because it is fixed to the constant framerate by the condition that one GOP is packed into a fixed number of BOPs or vice versa. In order to reduce the additional delay, we would have to increase the BOP rate $R_{\mathrm{BOP}}$, i.e. to decrease the size (in bytes) of the BOPs. Note that for a constant channel data rate $R$ this implies reducing the number $n$ of packets per BOP and/or the packet size $S$. Reducing $n$ results in a reduced efficiency of the RS channel coding scheme. Reducing the packet size reduces the transmission efficiency due to the overhead introduced by the packet headers. Since we focus mainly on Internet video streaming applications, where the transmission delay conditions are time varying as well as user dependent, a detailed delay optimization for a specific connection

is beyond the scope of this paper.

### 4.3 Error Concealment

If less than $k_l$ packets of a BOP are received, the RS erasure decoder cannot reconstruct the lost packets and the video decoder for layer $l$ receives a corrupted bitstream. In this case we apply a simple concealment strategy.

In the EL, missing image parts are replaced by spatially interpolated image content from the BL. If there is no image in the BL at this time instant, the missing image parts are copied from the last decoded EL image.

In the BL we employ simple temporal concealment, i.e. lost image parts are copied from the last decoded image of this layer. Since the motion information in the EL is predicted from the BL, losses in the BL can severely distort the EL. Therefore, we switch off the EL in case of losses in the BL. The EL is switched on again at a new picture start if there were no losses in the BL for this time instant.

## 5 Analytical PSNR Estimation

In this section we derive a theoretical framework which relates the perceived picture quality in terms of *PSNR* at the receiver to the packetization parameters and to the packet loss behavior of the transmission channel. This framework can be used to find appropriate packetization parameters for a gracefully degrading video transmission scheme instead of performing time-consuming transmission simulations.

To evaluate the performance of an $(n, k)$ RS code we need to know the probability that more than $n-k$ packets are lost since then the missing video packets cannot be reconstructed. We can compute this probability if we know the probability $P(m, n)$ of $m$ lost packets within the block of $n$ packets. $P(m, n)$ is called the *block error density function*. It is a simple binomial distribution in case of a memoryless channel with given packet loss probability. For our two-state Markov packet loss model described in section 2 we calculate $P(m, n)$ following the derivation given in [30].

The Markov model is a renewal model, i.e. the event of a loss frees the memory of the loss process and re-starts it. Such models are determined by the distribution of error-free intervals (gaps). Let a gap of length $\nu$ be the event that after a lost packet $\nu - 1$ packets are received and then again a packet is lost. The gap density function $g(\nu)$ gives the probability of a gap length $\nu$, i.e. $g(\nu) = \Pr(0^{\nu-1}1|1)$, where '1' denotes a lost packet, and '$0^{\nu-1}$' denotes $\nu - 1$ consecutively received packets. The gap distribution function $G(\nu)$ gives the probability of a gap length greater than $\nu - 1$, i.e. $G(\nu) = \Pr(0^{\nu-1}|1)$. In state B all packets are lost ('1'), while in state G all packets are received ('0'), yielding

$$g(\nu) = \begin{cases} 1 - p_{\mathrm{BG}} & \text{for } \nu = 1\,, \\ p_{\mathrm{BG}}\,(1 - p_{\mathrm{GB}})^{\nu-2} p_{\mathrm{GB}} & \text{for } \nu > 1\,, \end{cases}$$

$$G(\nu) = \begin{cases} 1 & \text{for } \nu = 1\,, \\ p_{\mathrm{BG}}\,(1 - p_{\mathrm{GB}})^{\nu-2} & \text{for } \nu > 1\,. \end{cases}$$

Let $R(m, n)$ be the probability of $m - 1$ packet losses within the next $n - 1$ packets following

a lost packet. It can be calculated using the recurrence

$$R(m,n) = \begin{cases} G(n) & \text{for } m = 1\,, \\ \sum\limits_{\nu=1}^{n-m+1} g(\nu) R(m-1, n-\nu) & \text{for } 2 \le m \le n\,. \end{cases}$$

Then the probability of $m$ lost packets within a block of $n$ packets is

$$P(m,n) = \begin{cases} \sum\limits_{\nu=1}^{n-m+1} P_{\mathrm{B}} G(\nu) R(m, n-\nu+1) & \text{for } 1 \le m \le n\,, \\ 1 - \sum\limits_{m=1}^{n} P(m,n) & \text{for } m = 0\,, \end{cases}$$

where $P_{\mathrm{B}}$ is the average loss probability.

Fig. 10 compares measured and calculated results for the block error density function $P(m,n)$ and the gap distribution function $G(\nu)$ for the transmission experiment described in Fig. 1. As can be seen, the model describes the packet channel's burst loss behavior very well. If the simple two-state model would not suffice to reproduce the measured $P(m,n)$, more sophisticated models as described in [31] could be employed.

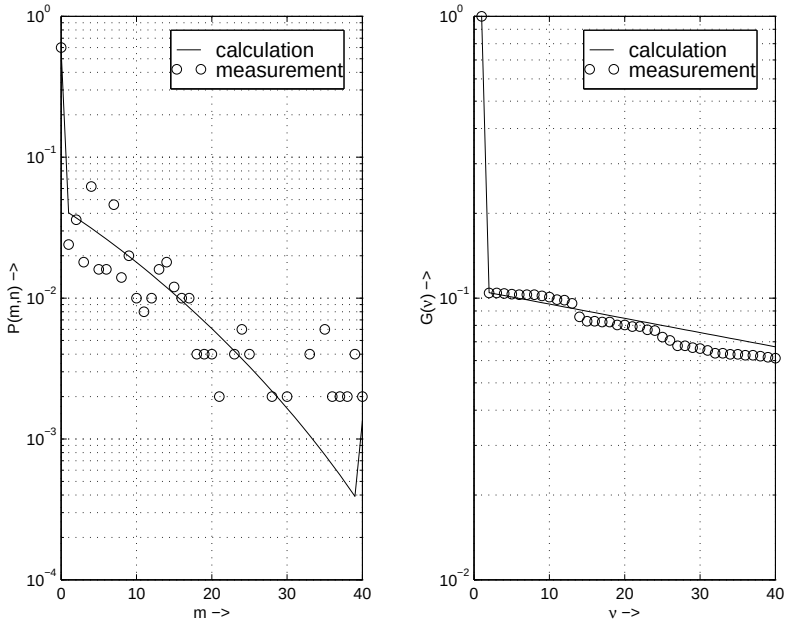


Figure 10: Block error density function $P(m,n)$ *(on the left)* and gap distribution function $G(\nu)$ *(on the right)* of the transmission Erlangen–Stanford (Fig. 1). Block length $n = 40$. 20000 packets or 500 blocks were transmitted. The channel parameters for the calculation were $P_{\mathrm{B}} = 0.0997$ and $L_{\mathrm{B}} = 9.57$.

Using $P(m,n)$, we can analytically calculate the residual loss probabilities experienced by a video decoder after RS decoding. If it is known how many losses are acceptable for the video decoder, this can be used to design the overall system. However, in a layered video coder the impact of the residual loss probabilities for the layers on the image quality is not obvious since it also depends on the quality difference between the layers. Therefore, we go one step further and estimate the resulting average *PSNR* of the proposed transmission scheme analytically.

Let us assume that layer $l$ is 'switched off' as soon as the video decoder detects a residual packet loss in this layer. In a scheme with $L$ layers $l = 0, \ldots, L-1$, each one protected with an $(n,k_l)$ RS code, layer $l$ is the highest resolution layer displayed if for the number of lost packets $m$ in the BOP $n - k_{l-1} < m \leq n - k_l$ holds, i.e. some data for layer $l-1$ is missing but not for layer $l$. Thus, the probability $p_l$ that layer $l$ is displayed can be computed as

$$p_l = \sum_{m=n-k_{l-1}+1}^{n-k_l} P(m,n)\,, \tag{3}$$

where we set formally $k_{-1} = n+1$ and $k_L = 0$. With $k_L = 0$ (3) yields the probability $p_L$ that even the base layer (BL) $L-1$ is affected by residual packet loss.

Let $PSNR_l$ be the average *PSNR* when layer $l$ is the highest resolution layer displayed over the whole time period encoded in one BOP, and let especially $PSNR_L$ be the *PSNR* if there are residual packet losses in the BL which is not switched off as described in section 4.3. $PSNR_L$ depends very much on the duration of the gap. We approximated it by measuring the *PSNR* between one decoded QCIF frame and all original frames of the test sequence. Note that all $PSNR_l$ can be measured at the encoder in advance. Then the average *PSNR* resulting from the switch-off strategy described above is

$$PSNR = \sum_{l=0}^{L} PSNR_l \cdot p_l\,. \tag{4}$$

If the layers are not completely switched off for the rest of a GOP but error concealment is applied, the average *PSNR* should be better than the one resulting from (4). This effect can be seen in Fig. 11 where we compare measured and calculated values for *PSNR* as a function of the average packet loss probability $P_{\mathrm{B}}$. The average *PSNR* with concealment is around 1 dB better than predicted. Yet, the *PSNR* estimation clearly shows the impact of the code parameters on the average *PSNR* under changing channel conditions, which will be used in the following to select the amount of error protection applied to each layer of our transmission scheme.

The video data rates $V_l$ are related to the channel data rate $R$ by (2). For given $n$, $V_l$, and $R$ we can vary the values of $k_l$, which should satisfy $k_{L-1} < \ldots < k_0 \leq n$ in order to ensure that more important data is protected better than less important data. A value of $k_0$ close to $n$ results in a low protection level for layer 0 (the EL) and reduces the average *PSNR* on channels with small packet loss probability. On the other hand, the lower resolution layers are protected better which results in a higher average *PSNR* on channels with high packet loss probability. The average *PSNR* for every set of $k_l$ can be calculated according to (4), and the one resulting in the desired behavior is selected.

An example for our two-layer scheme is shown in Fig. 12. The video data rates are $V_0 = 212$ kbps and $V_1 = 79$ kbps. We allow roughly 15% channel coding redundancy, resulting in a channel rate $R = 343$ kbps. In our simulations we always pack one GOP (Fig. 5) into one BOP (Fig. 9), which thereby contains 8 frames of a sequence encoded at 30 fps, resulting in a BOP rate $R_{\mathrm{BOP}} = 30\ \mathrm{s}^{-1}/8 = 3.75\ \mathrm{s}^{-1}$. During the transmission experiments described in section 1 we found that at packet rates $R_{\mathrm{P}} = nR_{\mathrm{BOP}}$ beyond $400\ \mathrm{s}^{-1}$ packet losses increased dramatically. To avoid this situation, the packet rate should fulfill $R_{\mathrm{P}} < 400\ \mathrm{s}^{-1}$. This constraint limits $n$ to $n < 400/3.75 = 107$. Since the RS coding efficiency increases with $n$, $n$ should be chosen as large as possible. Hence in the following we selected $n = 100$, which

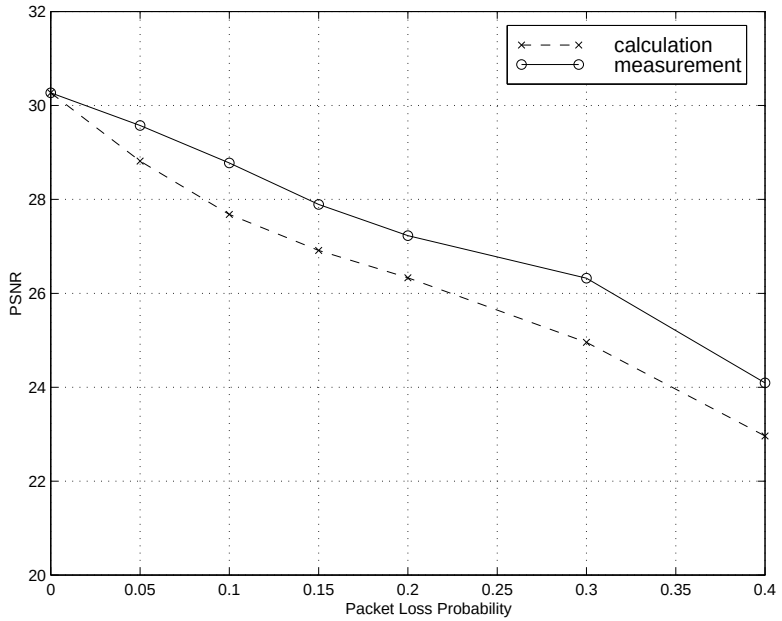


Figure 11: Measured and calculated average *PSNR* for *Silent Voice* as a function of the average packet loss probability $P_{\mathrm{B}}$. The average burst length is $L_{\mathrm{B}} = 5$ in this example. A scalable coder with 2 layers combined with UEP is used. For protecting the QCIF layer an $(100, 65)$ RS code, for protecting the CIF layer an $(100, 96)$ RS code is employed. The measured layer *PSNR*s are $PSNR_0 = 30.3$ dB (CIF), $PSNR_1 = 26.6$ dB (QCIF), and $PSNR_2 = 21$ dB.

results in an average packet size of $S = 114$ bytes. In Fig. 12, $k_0$ is varied between 100 and 85. $k_0 = k_1 = 85$ marks the equal error protection case, which performs best at low packet loss probabilities $P_{\mathrm{B}}$, while the case with unprotected layer 0 ($k_0 = 100$) performs best at high $P_{\mathrm{B}}$. For the simulations presented in the following section we selected the case $k_0 = 96$, $k_1 = 65$, which compared to the other cases results in an only slightly decreased *PSNR* at low $P_{\mathrm{B}}$ but large *PSNR* improvements at high $P_{\mathrm{B}}$.

# 6 Simulation Results

In this section we present simulation results obtained with the transmission scheme presented in section 4. We compare the performance of the proposed scheme with a single layer H.263 encoded video stream transmitted without protection and then transmitted with forward error correction.

All experiments have been carried out without employing any bit rate control method. Instead fixed quantizer step sizes were used. For the single layer coding experiments we simply adjusted the step sizes such that the desired overall source bit rate was met. For the two-layer pyramid we distributed the overall available source bit rate to the base and the enhancement layer according to the equal noise bit rate allocation rule derived in [32]. This rule simply states that if we encode the pictures in the QCIF and the CIF layer at roughly the same PSNR, we are sufficiently close to the optimum bit rate allocation for a two-layer spatial pyramid coder.

The input sequences for the experiments in this section are the CIF test sequences *Silent*

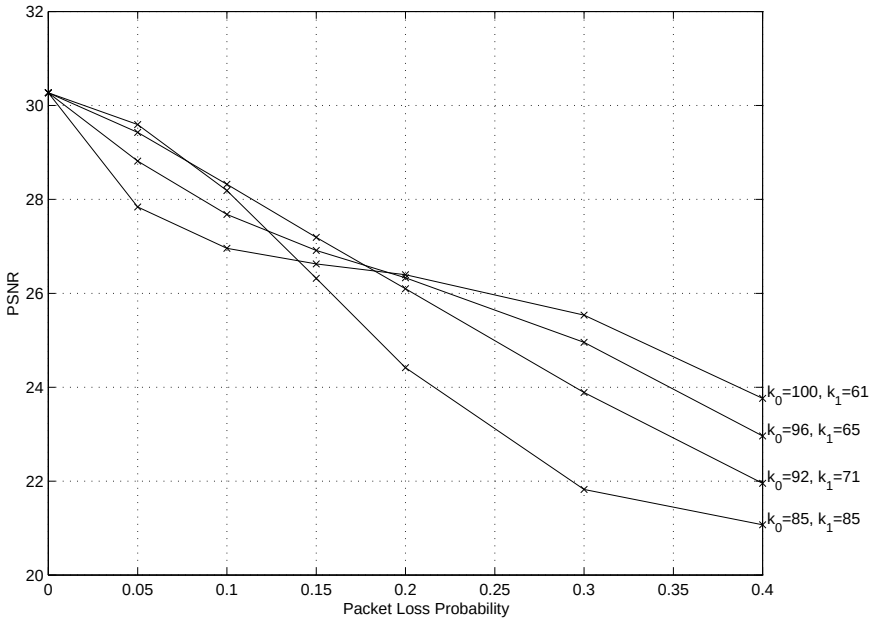


Figure 12: Calculated average *PSNR* for *Silent Voice* vs. average packet loss probability $P_{\mathrm{B}}$. Average burst length $L_{\mathrm{B}} = 5$. Number of packets per BOP $n = 100$. Pyramid Coder with 2 layers and UEP. The layer *PSNR*s are $PSNR_0 = 30.3$ dB (CIF), $PSNR_1 = 26.6$ dB (QCIF), and $PSNR_2 = 21$ dB. $k_0$ and $k_1$ are varied for a fixed average channel code rate of 0.85.

*Voice* and *Foreman* encoded at 30 fps. We allow 15 % redundancy for error protection. Table 1 gives an overview about the bit rates and the error protection parameters we used in our experiments.

Figs. 13 and 14 show the averaged *PSNR* over 5 channel realizations as a function of the packet loss probability $P_{\mathrm{B}}$. It can be seen that the proposed transmission scheme which combines a scalable coder with unequal error protection achieves graceful degradation. In the error free case unprotected H.263 coding is the best choice as expected since all available bits can be spent for the video source encoding. However, the quality drops very rapidly resulting in very annoying distortions even if only a few packets are lost. Applying FEC to the single layer H.263 coder just shifts the break down point to higher packet loss rates. However, as discussed below, the subjective quality has already degraded significantly at this point. In contrast to the single layer coding schemes, the scalable scheme can decode an almost error-free base layer, even if parts of the enhancement layer are lost. This results in a reduced spatial and temporal resolution but the annoying effects observed in the single layer cases are significantly reduced.

For the test sequence *Foreman*, Fig. 15 compares for the test sequence *Foreman PSNR* curves for a single channel realization with $P_{\mathrm{B}} = 0.15$ obtained with the scalable coder combined with UEP with those obtained with the single layer H.263 coder combined with EEP. Evidently the scalable coder switches to the base layer very often since the enhancement layer is not protected strongly enough for this channel. In contrast, for H.263 the *PSNR* stays at the CIF quality level most of the time but drops much deeper from time to time resulting in severely distorted images until the next Intra-frame is received.

Figs. 16 and 17 give an impression of the subjective picture quality obtained with the

Table 1: This table summarizes the simulation experiments we carried out for the two test sequences (*Silent Voice* and *Foreman*). In each of the listed experiments we measured the resulting PSNR as a function of the packet loss probability. The corresponding graphs are presented in Figs. 13 and 14.

| Experiment | Source rate [kbps] | Error protection | Channel rate [kbps] |
|---|---|---|---|
| Test sequence: Silent Voice | | | |
| Scalable coder | 290 | QCIF: $(100, 65)$ RS, CIF: $(100, 96)$ RS | 340 |
| H.263 protected | 290 | $(100, 85)$ RS | 340 |
| H.263 unprotected | 340 | – | 340 |
| Test sequence: Foreman | | | |
| Scalable coder | 410 | QCIF: $(100, 62)$ RS, CIF: $(100, 96)$ RS | 480 |
| H.263 protected | 410 | $(100, 85)$ RS | 480 |
| H.263 unprotected | 480 | – | 480 |

scalable coder and with the error protected single layer H.263 coder.

Although most of the software modules have not been optimized for speed, the decoding process for all presented experiments including RS channel decoding, source decoding and display of the decoded pictures runs in real-time on a 166 MHz Sun UltraSPARC 2 workstation.

# 7 Conclusions

In this article we presented a transmission scheme for Internet video streaming which degrades gracefully as the packet loss probability of an Internet connection increases. The transmission scheme consists of a two-layer scalable video coder which employs a fully standard compatible H.263 coder in its base layer, complemented by an enhancement layer encoder which uses an $E_8$-lattice vector quantizer. We described how the scalable video coder can be combined with an unequal error protection scheme based on Reed–Solomon codes. We reviewed an analytically tractable two-state Markov model to simulate the typical Internet packet loss behavior. A fast and flexible implementation of a Reed–Solomon channel coder and decoder based on Newton's interpolation was described. We derived a theoretical framework for the overall system by which the Internet packet loss behavior can be related to the picture quality perceived at the receiver. This framework was used to select reasonable parameter settings for our experiments. The results of those experiments show that the presented system achieves a gracefully degrading picture quality for packet losses up to 30%. The proposed scheme is especially useful for Internet multicast applications where different users are perceiving different channel qualities and no feedback channel can be employed. No specific support from the network layer is required.

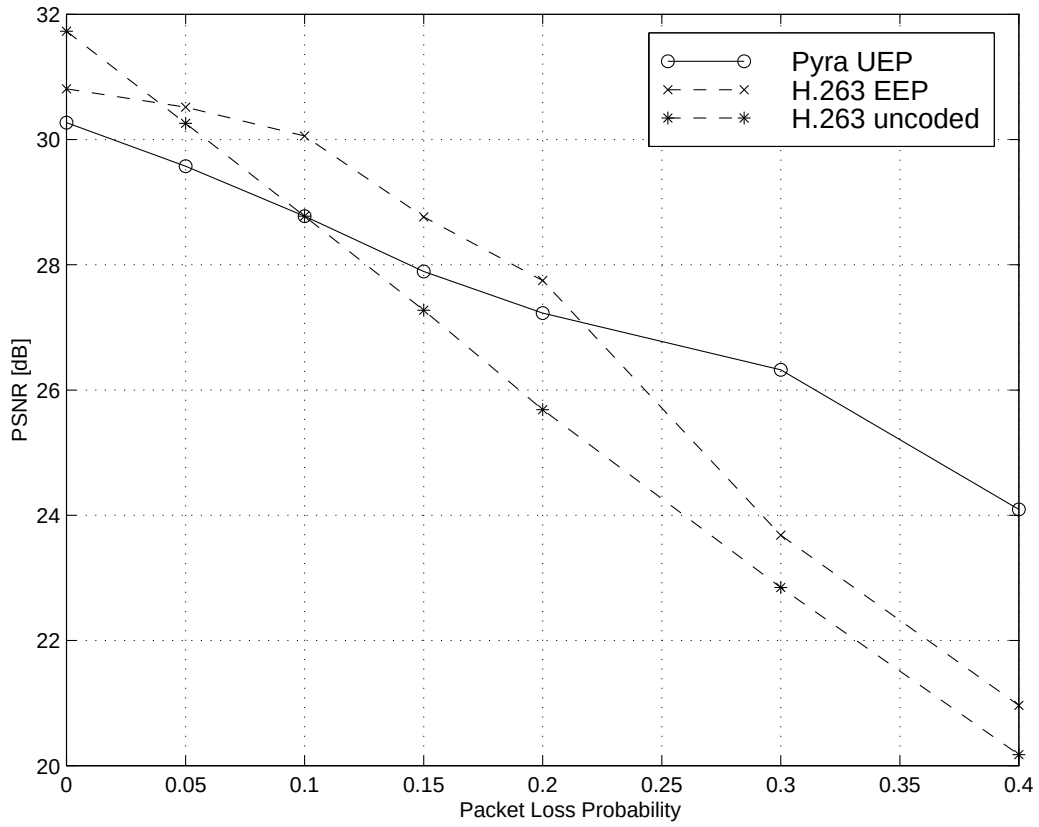


Figure 13: Average *PSNR* for *Silent Voice*. H.263 without error protection, H.263 with EEP $(100, 85)$ RS and the Pyramid Coder with 2 layers and UEP QCIF: $(100, 65)$ RS, CIF: $(100, 96)$ RS. Average burst length $L_{\mathrm{B}} = 5$.

# References


[1] G. Fleming, A. E. Hoiydi, J. De Vriendt, G. Nikolaidis, Piolini F., and M. Maraki. A flexible network architecture for UMTS. *IEEE Personal Communications*, pages 8–15, Apr. 1998.

[2] B. Kreller, A. S.-B. Park, J. Meggers, G. Forsgren, E. Kovacs, and M. Rosinus. UMTS: A middleware architecture and mobile API approach. *IEEE Personal Communications*, pages 32–38, Apr. 1998.

[3] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. IETF RFC 1889, Jan. 1996.

[4] G. Morrison and I. Parke. A spatially layered hierarchical approach to video coding. *Signal Processing: Image Communication*, 5(5-6):445–462, Dec. 1993.

[5] B. Girod. Scalable video for multimedia systems. *Computers & Graphics*, 17(3):269–276, 1993.

[6] N. Chadda, G. Wall, and B. Schmidt. An end-to-end software-only scalable video delivery system. In *Proc. NOSSDAV'95*, Apr. 1995.

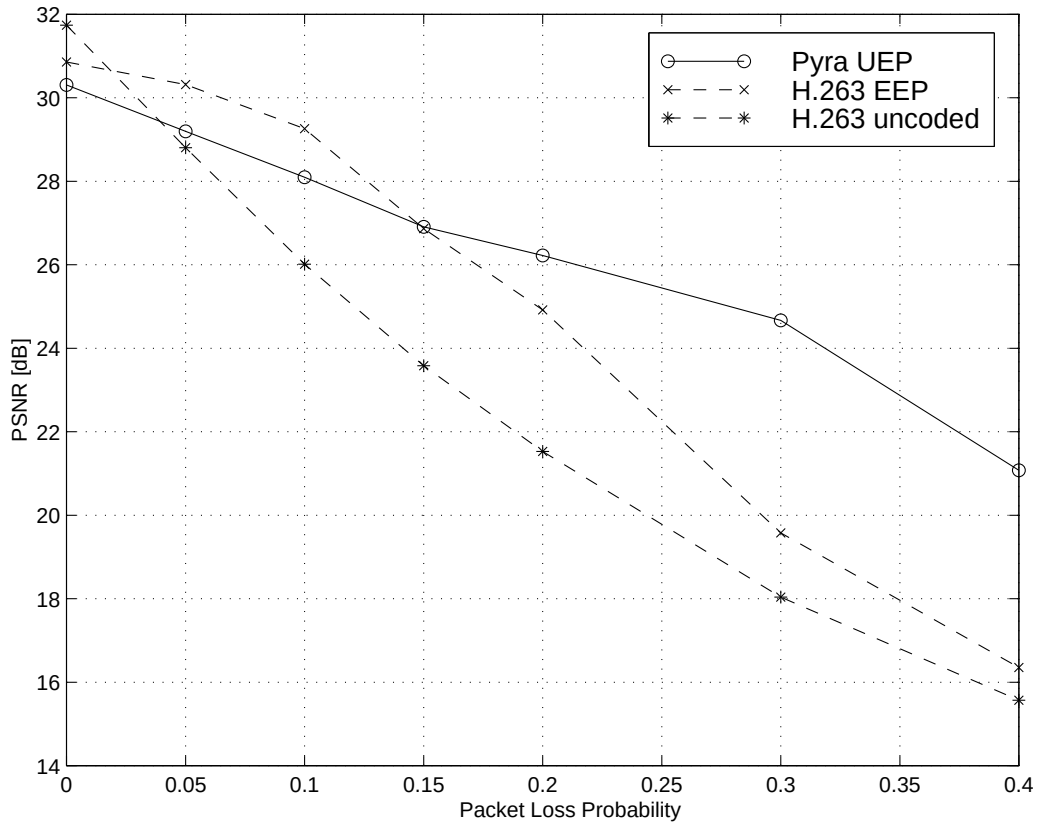


Figure 14: Average *PSNR* for *Foreman*. H.263 without error protection, H.263 with EEP $(100, 85)$ RS and the Pyramid Coder with 2 layers and UEP QCIF: $(100, 64)$ RS, CIF: $(100, 96)$ RS. Average burst length $L_{\mathrm{B}} = 5$.


[7] B. Girod, U. Horn, and B. Belzer. Scalable video coding with multiscale motion compensation and unequal error protection. In Y. Wang, S. Panwar, S.-P. Kim, and H. L. Bertoni, editors, *Multimedia Communications and Video Coding*, pages 475–482. Plenum Press, New York, Oct. 1996.

[8] D. Hoffman and M. Speer. Hierarchical video distribution over Internet-style networks. In *Proc. ICIP'96*, volume I, pages 5–8, Lausanne, Sep. 1996.

[9] W. Tan, E. Chang, and A. Zakhor. Real time software implementation of scalable video codec. In *ICIP'96*, volume I, pages 17–20, Lausanne, Sep. 1996.

[10] R. Mathew and J. F. Arnold. Layered coding using bitstream decomposition with drift correction. *IEEE Trans. on Circuits and Systems for Video Technology*, 7(6):882–891, Dec. 1997.

[11] U. Horn and B. Girod. Scalable video transmission for the Internet. *Computer Network and ISDN Systems*, 29(15):1833–1842, Nov. 1997.

[12] U. Horn, B. Girod, and B. Belzer. Scalable video coding for multimedia applications and robust transmission over wireless channels. In *7th Int. Workshop on Packet Video*, pages 43–48, Brisbane, Mar. 1996.

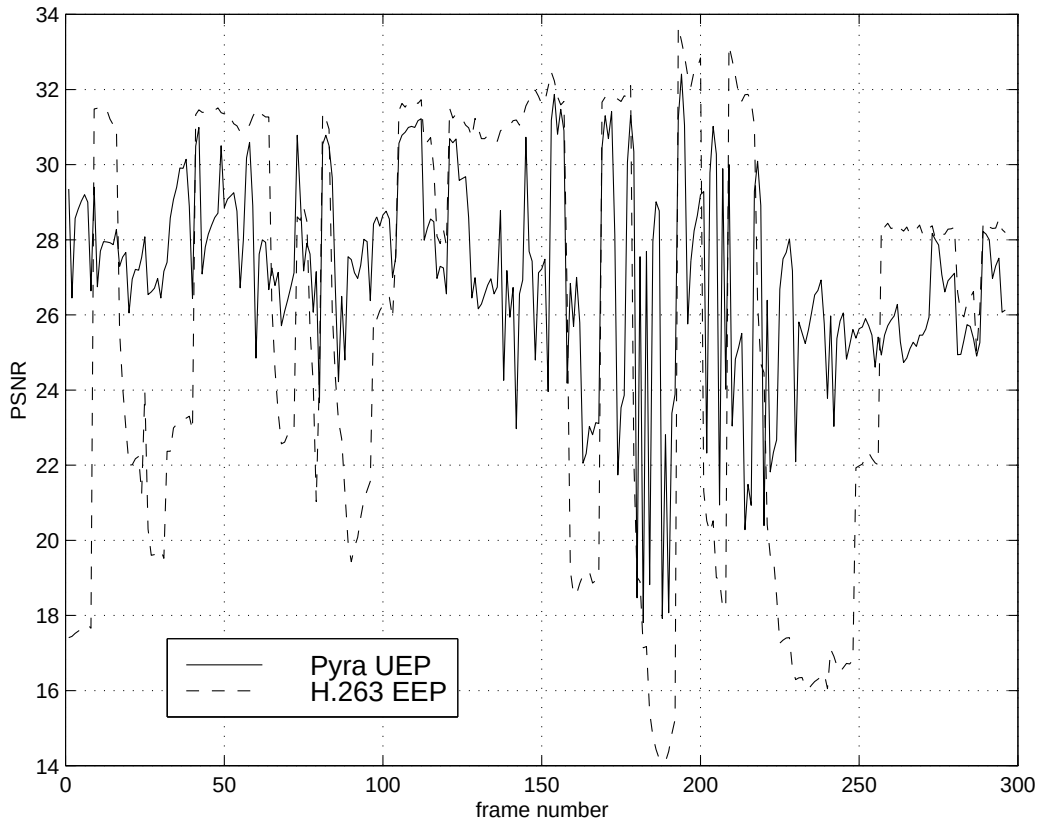


Figure 15: *PSNR* for *Foreman*. Pyramid Coder with 2 layers and UEP $(100, 64)$ RS, $(100, 96)$ RS and H.263 with EEP $(100, 85)$ RS. One channel realization with packet loss probability $P_{\mathrm{B}} = 0.15$ and average burst length $L_{\mathrm{B}} = 5$.

[13] A. Albanese, J. Bloemer, J. Edmonds, M. Luby, and M. Sudan. Priority encoding transmission. *IEEE Trans. on Inf. Theory*, 42(6):1737–1747, Nov. 1996.

[14] C. Leicher. Hierarchical encoding of MPEG sequences using priority encoding transmission (PET). Technical Report TR-94-058, International Computer Science Institute, Berkeley, Nov. 1994.

[15] S. H. Lee, P. J. Lee, and R. Ansari. Cell loss detection and recovery in variable rate video. In *Proc. 3rd Int. Packet Video Works.*, Morriston, March 1990.

[16] E. W. Biersack. Performance evaluation of forward error correction in an ATM environment. *IEEE J. on Sel. Areas in Com.*, 11(4):631–640, May 1993.

[17] V. Parthsarathy, J. W. Modestino, and K. S. Vastola. Design of a transport coding scheme for high–quality video over ATM networks. *IEEE Trans. Circ. Syst. Video Techn.*, 7(2):358–376, Apr. 1997.

[18] V. Kumar. *MBone - Interactive Multimedia on the Internet.* New Riders Publishing, Indianapolis, 1996.

Figure 16: One decoded image of the Pyramid Coder.


[19] R. E. Blahut. *Theory and Practice of Error Control Codes.* Addison Wesley, Reading, MA, 1983.

[20] A. J. McAuley. Reliable broadband communication using a burst erasure correcting code. In *Proc. ACM SIGCOMM*, pages 297–306, Philadelphia, Sep. 1990.

[21] U. K. Sorger. A new Reed–Solomon decoding algorithm based on Newton's interpolation. *IEEE Trans. Inform. Theory*, 39:358–365, Mar. 1993.

[22] M.K. Uz, M. Vetterli, and D.J. LeGall. Interpolative multiresolution coding of advanced television with compatible subchannels. *IEEE Trans. on Circ. and Syst. for Video Tech.*, 1(1):86–99, Mar. 1991.

[23] G. J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Sig. Proc. Mag.*, November 1998.

[24] U. Horn and B. Girod. Estimation of motion vector fields for multiscale motion compensation. In *ITG-Fachbericht 143 - Proc. PCS'97*, pages 141–144, Berlin, Sep. 1997. VDE Verlag.

[25] J. H. Conway and N. J. A. Sloane. *Sphere Packings, Lattices and Groups.* Springer, 1988.

[26] J. H. Conway and N. J. A. Sloane. Fast quantizing and decoding algorithms for lattice quantizers and codes. *IEEE Trans. on Inf. Theory*, IT-28(2):227–232, Mar. 1982.

[27] J. H. Conway and N. J. A. Sloane. A fast encoding method for lattice codes and quantizers. *IEEE Trans. on Inf. Th.*, 29(6):820–824, Nov. 1983.

[28] ITU-T. Rec. H.263, Video Coding for Low Bitrate Communication, Version 2, Jan. 1998.

Figure 17: One decoded image of H.263 with EEP.

[29] A. R. Reibmann and B. G. Haskell. Constraints on variable bit-rate video for atm networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 2(4):361–372, Dec. 1992.

[30] E. O. Elliott. A model of the switched telephone network for data communications. *Bell. Syst. Techn. J.*, 44:89–109, Jan. 1965.

[31] L. N. Kanal and A. R. K. Sastry. Models for channels with memory and their applications to error control. *Proc. IEEE*, 66(7):724–744, Jul. 1978.

[32] U. Horn, T. Wiegand, and B. Girod. Bit allocation methods for closed-loop coding of oversampled pyramid decompositions. In *Proc. ICIP'97*, volume II, pages 17–20, Santa Barbara, Okt. 1997.