# Compressed Video Over Networks

Editors: Ming-Tin Sun and Amy R. Reibman

# Chapter 12: Wireless Video

Bernd Girod and Niko Färber

Telecommunications Laboratory
University of Erlangen-Nuremberg
Cauerstrasse 7, 91058 Erlangen, Germany
Phone: +49 9131 8527100
Fax: +49 9131 8528849
girod@LNT.de
faerber@LNT.de

Nov. 14, 1999

# 1 Introduction

In the last decade, both mobile communications and multimedia communications have experienced unequaled rapid growth and commercial success. Naturally, the great – albeit separate – successes in both areas fuel the old vision of ubiquitous multimedia communication – being able to communicate from anywhere at any time with any type of data. The convergence of mobile and multimedia is now underway. Building on advances in network infrastructure, low-power integrated circuits, and powerful signal processing/compression algorithms, wireless multimedia services will likely find widespread acceptance in the next decade. The goals of current second-generation cellular and cordless communications standards – supporting integrated voice and data – are being expanded in third-generation wireless networks to provide truly ubiquitous access and integrated multimedia services. This vision is shared by many, e.g., by Ericsson's GSM pioneer Jan Uddenfeldt when he writes *"The tremendous growth of Internet usage is the main driver for third-generation wireless. Text, audio, and image (also moving) will be the natural content, i.e., multimedia, for the user."* [1].

Video communication is an indispensable modality of multimedia, most prominently exemplified by the Internet-based World Wide Web today. After the Web browser itself, audio/video streaming decoders have been the most frequently downloaded Internet application, and they will be part of the browser software by the time this chapter appears in print. Real-time audiovisual communication will also be an integral part of third-generation wireless communication services. The current vision includes a small handheld device that allows the user to communicate from anywhere in the world with anyone in a variety of formats (voice, data, image, and full-motion video) from virtually any geographic location. This next generation of wireless multimedia communicators is expected to be equipped with a camera, a microphone, and a liquid crystal color display, serving both as a videophone and computer screen. The conventional lap-top keyboard is likely to be replaced by a writing tablet, facilitating optical handwriting recognition and signature verification. With progressing miniaturization of components, wristwatch "Dick Tracy" communicators are expected to follow soon after.

Of all modalities desirable for future mobile multimedia systems, motion video is the most demanding in terms of bit-rate, and is hence likely to have the strongest impact on network architecture and protocols. Even with state-of-the-art compression, television quality requires a few Megabits per second (Mbps), while for low-resolution, limited-motion video sequences, as typically encoded for picturephones, a few tens of kbps are required for satisfactory picture quality [2]. Today's "second-generation" cellular telephony networks, such as Global System for Mobile Communications (GSM), typically provide 10 - 15 kbps, suitable for compressed speech, but too little for motion video. Fortunately, the standardization of higher-bandwidth networks, such as Universal Mobile Telecommunications System (UMTS) [3] [4], is well underway, and, together with continued progress in video compression technology, wireless multimedia communicators with picturephone functionality and Internet videoserver access will be possible.

Beyond the limited available bit-rate, wireless multimedia transmission offers a number of interesting technical challenges. A recent review has appeared in [5]. One of the more difficult issues is due to the fact that mobile networks cannot provide a guaranteed quality of service, because high bit error rates occur during fading periods. Transmission errors of a mobile wireless radio channel range from single bit errors to burst errors or even an intermittent loss of the connection. The classic technique to combat transmission errors is Forward Error Correction (FEC), but its effectiveness is limited due to widely varying error conditions. A worst case design would lead to a prohibitive amount of redundancy. Closed-loop error control techniques like Automatic Repeat reQuest (ARQ) [6] have been shown to be more effective than FEC and successfully applied to wireless video transmission [7] [8]. Retransmission of corrupted data frames, however, introduces additional delay, which might be unacceptable for real-time conversational or interactive services. As a result, transmission errors cannot be avoided with a mobile radio channel, even when FEC and ARQ are combined. Therefore, the design of a wireless video system always involves a trade-off between channel coding redundancy that protects the bit-stream and source coding redundancy deliberately introduced for greater error resilience of the video decoder.

Without special measures, compressed video signals are extremely vulnerable against transmission errors. Basically, every bit counts. Considering specifically low bit-rate video, compression schemes rely on interframe coding for high coding efficiency, i.e., they use the previous encoded and reconstructed video frame to predict the next frame. Therefore, the loss of information in one frame has considerable impact on the quality of the following frames. Since some residual transmission errors will inevitably corrupt the video bit-stream, this vulnerability precludes the use of low bit-rate video coding schemes designed for error-free channels without special measures. These measures have to be built into the video coding and decoding algorithms themselves and form the "last line of defense" if techniques like FEC and ARQ fail.

A comprehensive review of the great variety of error control and concealment techniques that have been proposed during the last 10 - 15 years has been presented in an excellent paper by Wang and Zhu recently [9] and is also included in Chapter 6 of this book. For example, one can partition the bit-stream into classes of different error sensitivity (often referred to as data partitioning) to enable the use of unequal error protection [10] [11] [12]. Data partitioning has been included as an error resilience tool in the MPEG-4 standard [13]. Unequal error protection can significantly increase the robustness of the transmission and provide graceful degradation of the picture quality in case of a deteriorating channel. Since unequal error protection does not incorporate information about the current state of the mobile channel, the design of such a scheme is a compromise that accommodates a range of operating conditions. Feedback-based techniques, on the other hand, can adjust to the varying transmission conditions rapidly and make more effective use of the channel. This leads us to the notion of *channel-adaptive source coding*.

The ITU-T Study Group 16 has adopted feedback-based error control in their effort towards mobile extensions of the successful Recommendation H.263 (see Chapter 1, "H-Series Video Coding Standards") for low bit-rate video coding. The first version of H.263 already included *Error Tracking*, a technique that allows the encoder to accurately estimate interframe error propagation and adapt its encoding strategy to mitigate the effects of past transmission errors [14] [15]. The second version, informally known as H.263+, was adopted by the ITU-T in February 1998. Among many other enhancements, it contains two new optional modes supporting Reference Picture Selection (Annex N) and Independent Segment Decoding (Annex R) as an error confinement technique [16] [17]. Additional enhancements, for example, data partitioning, unequal error protection, and reversible variable length coding, are under consideration for future versions of the standard, informally known as H.263++ and H.26L.

Most of the error control schemes for wireless video are pragmatic engineering solutions to a problem at hand that do not generalize. The trade-offs in designing the overall transmission chain are not well-understood and need further study that ultimately should lead to a general theoretical framework for joint optimization of source coding, channel coding and transport protocols, coding schemes with superior robustness and adaptability to adverse transmission condition, and multimedia-aware transport protocols that make most efficient use of limited wireless network resources. In the meantime, we have to be content with more modest goals.

In this chapter, we investigate the performance and trade-offs when using established error control techniques for wireless video. We set the stage by discussing the basic trade-off between source and channel coding redundancy in Section 2 and introduce the distortion-distortion function as a formal tool for comparing wireless video systems. In Section 3, we briefly discuss how to combat transmission errors by channel coding and illustrate the problems that are encountered with classic FEC applied to a fading channel. We also discuss error amplification that can occur with IP packetization over wireless channels. In Section 3, we discuss error resilience techniques for low bit-rate video, with particular emphasis on techniques adopted by the ITU-T as part of the H.263 Recommendation. These techniques include feedback-based error control, yielding in effect a channel-adaptive H.263 encoder. The various approaches are compared by means of their operational distortion-distortion function under the same experimental conditions.

# 2 Trading Off Source and Channel Coding

Naturally, the problem of transmitting video over noisy channels involves both source and channel coding. The classic goal of source coding is to achieve the lowest possible distortion for a given target bit-rate. This goal has a fundamental limit in the rate-distortion bound for given source statistics. The source-coded bitstream then needs to be transmitted reliably over a noisy channel. Similar to the rate-distortion bound in source coding, the channel capacity quantifies the maximum rate at which information can be transmitted reliably over the given channel. Hence, the classic goal of channel coding is to deliver reliable information at a rate that is as close as possible to the channel capacity. According to Shannon's *Separation Principle*, it is possible to independently consider source and channel coding without loss in performance [18]. However, this important information-theoretic result is based on several assumptions that might break down in practice. In particular, it is based on (1) the assumption of an infinite block length for both source and channel coding (and hence infinite delay) and (2) an exact and complete knowledge of the statistics of the transmission channel. As a corollary of (2), the Separation Principle applies only to point-to-point communications and is not valid for multiuser or broadcast scenarios [18]. Therefore, *Joint Source-Channel Coding* (JSC coding) can be advantageous in practice.

A joint optimization of source and channel coding can be achieved by exploiting the redundancy in the source signal for channel decoding (*source-controlled channel decoding*, e.g., [19]) or by designing the source codec for a given channel characteristic (*channel-optimized source coding*, e.g., [20]). In either case, source and channel coding can hardly be separated anymore and are truly optimized jointly. Unfortunately, joint source-channel coding schemes for video are in their infancy today. A pragmatic approach for today's state-of-the-art is to keep the source coder and the channel coder separate, but optimize their parameters jointly. This approach will be followed in this chapter. A key problem of this optimization is the bit allocation between the source and channel coder that will be discussed below. To illustrate the problem we first consider the typical components of a wireless video system. For more information on separate, concatenated, and joint source-channel coding for wireless video see [21].

## 2.1 Components of a Wireless Video System

Fig. 1 shows the basic components of a wireless video system. The space-time discrete input video signal $i[x, y, t]$ is fed into a video encoder. The video encoder is characterized by its operational distortion–rate function $D_e(R_e)$, where $e[x, y, t]$ is the reconstructed video signal at the encoder and $R_e$, $D_e$ are the average rate and average distortion respectively. After source coding, the compressed video bitstream is prepared for transmission over a network. Often, this involves packetization. This is particularly the case for transmission employing the *Internet Protocol* (IP). The correct delivery of packets requires a multitude of functionalities that need to be provided by the network, such as routing, hand-over, packet fragmentation and reassembly, flow control, etc. These functionalities are not covered in this chapter (see Chapter 3, "IP Networks" instead), and, for now, we assume that the corresponding protocol layers are transparent and do not introduce losses. In practice, this assumption is not always justified, and we hence revisit this issue in Section 3.4

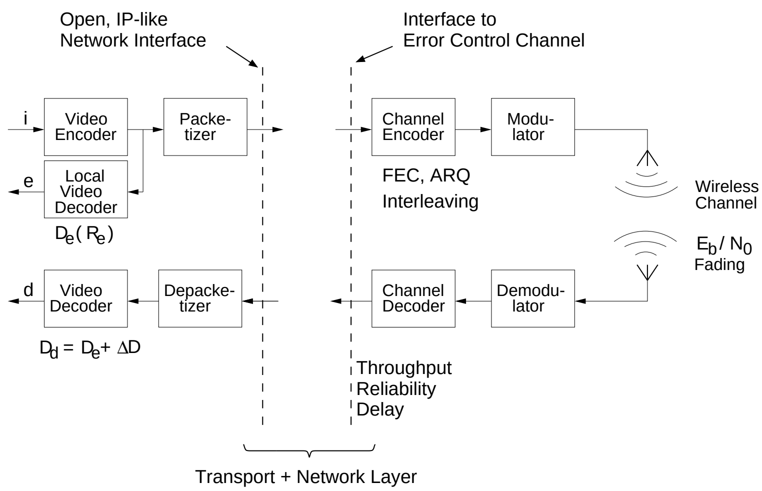


Figure 1: Basic components of a video transmission system.

In wireless video systems, the end-to-end transmission typically comprises one or two wireless radio extensions to a wired backbone, at the beginning and/or the end of a transmission chain. Therefore, the packetized bitstream is transmitted at least once over a wireless channel as illustrated in Fig. 1. In contrast to the wired backbone, the capacity of the wireless channel is fundamentally limited by the available bandwidth of the radio spectrum and various types of noise and interference. Therefore, the wireless channel can be regarded as the "weakest link" of future multimedia networks and, hence, requires special attention, especially if mobility gives rise to fading and error bursts. The resulting transmission errors introduced require *error control* techniques. A classic technique is FEC that can be combined with *Interleaving* to reduce the effect of burst errors. On the other hand, closed-loop error control techniques like ARQ are particularly attractive if the error conditions vary in a wide range. These error control techniques are part of the channel codec and are discussed in more detail in Section 3.

The bitstream produced by the channel encoder is represented by an analog signal waveform suitable for the transmission channel by the modulator. The power of the channel noise that is superimposed to the transmitted signal has to be evaluated with respect to the energy that is used for the transmission of each bit. Therefore, the ratio of bit-energy to noise-spectral-density ($E_b/N_0$) is often used to characterize the noisiness of the channel. Other parameters that describe the correlation of errors are also of importance. After demodulation, the channel decoder tries to recover from transmission errors by exploiting the error correction capability of the FEC scheme or by requesting a retransmission of corrupted data frames.

The term *Error Control Channel* refers to the combination of the channel codec, the modulator/demodulator and the physical channel [23]. Ideally, the error control channel would provide an error-free binary link with a guaranteed bit-rate and maximum delay to the video coder. However, as we will see in Section 3, the effectiveness of channel coding is limited in a mobile environment when data have to be transmitted with low delay. Essentially, only a compromise between (1) *reliability*, (2) *throughput*, and (3) *delay* can be achieved. This fundamental trade-off is typical for the communication over noisy channels and has to be considered for the design of wireless video systems.

Because the error control channel has to balance reliability, throughput, and delay, some residual transmission errors usually remain after channel decoding, especially for low-latency applications. In this case, the video decoder must be capable of processing an erroneous bitstream. The residual errors cause an additional distortion $\Delta D$ such that the decoded video signal $d[x,y,t]$ contains the total average distortion $D_d = D_e + \Delta D$.

## 2.2 Distortion Measures

For a quantitative analysis of wireless video systems, we require measures for the video signal distortion introduced by the source encoder ($D_e$) or the distortion at the output of the video decoder ($D_d$). Clearly, since the decoded video signal is ultimately played back to a human observer, a distortion measure should be consistent with the perceived subjective quality. In practice, the most common distortion measure for video coding is *Mean Squared Error* (MSE). Though MSE is notorious for its flaws as a measure of subjective picture quality, it provides consistent results as long as the video signals to be compared are affected by the same type of impairment [24]. For example, the subjective quality produced by a particular video codec at two different bit-rates for the same input signal can usually be compared by MSE measurements because both decoded signals contain similar quantization and blocking artifacts. Hence, we define the distortion at the encoder as

$$D_e = \frac{1}{XYT} \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{t=1}^{T} (i[x,y,t] - e[x,y,t])^2, \tag{1}$$

for a frame size of $X \times Y$ pixels and $T$ encoded video frames. If the distortion shall be calculated for individual frames, we can obtain $D_e[t]$ by calculating the MSE for each frame separately.

The obvious approach to measure the distortion at the decoder after transmission is to calculate the MSE between the received video signal $d[x,y,t]$ and the original video signal $i[x,y,t]$. In fact, this is frequently done in the literature to evaluate video transmission systems [25] [26] [27]. Due to the probabilistic nature of the channel, one has to consider the distortion averaged over many different realizations of the channel. For a particular constellation of the wireless video system (i.e., FEC rate, $E_b/N_0$, encoding parameters of video codec, ...), we therefore obtain decoded signals for each realization $l$, denoted as $d_l[x,y,t]$. Assuming $L$ realizations, the MSE at the decoder is then calculated as

$$D_d = \frac{1}{XYTL} \sum_{x=1}^{X} \sum_{y=1}^{Y} \sum_{t=1}^{T} \sum_{l=1}^{L} (i[x,y,t] - d_l[x,y,t])^2. \tag{2}$$

Sometimes it is necessary to also calculate the distortion $D_e$ at the encoder by averaging over many realizations of the channel, similar to (2). In particular, this is the case for channel-adaptive source coding where the operation of the encoder depends on the behavior of the channel as discussed in Section 4.

Note that two types of distortion appear in the decoded video signal $d$, i.e., the distortion due to source coding and the distortion caused by transmission errors. While the former is adequately described by $D_e$, we define

$$\Delta D = D_d - D_e, \tag{3}$$

to describe the latter. Typically, $D_e$ is the result of small quantization errors that are evenly distributed over all encoded frames, while $\Delta D$ is dominated by strong errors that are concentrated in a small part of the picture and are (hopefully!) present only for a short time. Because such errors are perceived very differently, an average measure such as $D_d$ alone can be misleading if not applied carefully. Instead, both $D_e$ and $\Delta D$ should be considered for the evaluation of video quality simultaneously as discussed in the next section.

Before concluding this section, we note that MSE is commonly converted to peak-signal-to-noise ratio (PSNR) in the video coding community. PSNR is defined as $10 \log_{10}(255^2/MSE)$, where 255 corresponds to the peak-to-peak range of the encoded and decoded video signal (each quantized to 256 levels). It is expressed in decibels (dB) and increases with increasing picture quality. Though the logarithmic scale provides a better correlation with subjective quality, the same limitations as for MSE apply. As a rule of thumb for low bit-rate video coding (with clearly visible distortions), a difference of 1 dB generally corresponds to a noticeable difference while acceptable picture quality requires values greater than 30 dB. Since PSNR is more commonly used than MSE, we will use instead of (1), (2), and (3)

$$PSNR_e = 10 \log_{10} \frac{255^2}{D_e}, \tag{4}$$

$$PSNR_d = 10 \log_{10} \frac{255^2}{D_d}, \text{and} \tag{5}$$

$$\Delta PSNR = PSNR_e - PSNR_d = 10 \log_{10} \frac{D_e}{D_d} = 10 \log_{10} \frac{D_e}{D_e + \Delta D}, \tag{6}$$

when presenting experimental results. Now, after having defined the necessary distortion measures, we return to the problem of bit allocation between source and channel coding by introducing the distortion-distortion function.

### 2.3 Distortion – Distortion Function

Consider again the wireless video transmission system illustrated in Fig. 1. Assume that a modulation scheme is used which provides a constant "raw" bit-rate $R_c$. By operating the video encoder at a bit-rate $R_e \leq R_c$, the remaining bit-rate $R_c - R_e$ can be utilized for error control information to increase the reliability of the transmission over the wireless channel and thus reduce the *Residual Word Error Rate* (RWER) which describes the probability of residual errors after channel decoding. As noted above, there is a fundamental trade-off between *throughput* and *reliability*, corresponding to the bit allocation between source and channel coding characterized by the code rate $r = R_e/R_c$.

Altering the bit allocation between source and channel coding has two effects on the picture quality of the video signal $d$ at the decoder output. First, a reduction of $r$ reduces the bit-rate available to the video encoder and thus degrades the picture quality at the encoder regardless of transmission errors. The actual $\text{PSNR}_e$ reduction is determined by the operational distortion-rate function $D_e(R_e)$ of the video encoder. On the other hand, the residual error rate is reduced when reducing $r$, determined by the properties of the error control channel, i.e., the channel codec, the modulation scheme, and the characteristic of the channel. Finally, a reduction in RWER leads to a reduction in $\Delta$PSNR depending on several implementation issues, such as resynchronization, packetization, and error concealment, all of which are associated with the video decoder. The interaction of the various characteristics are illustrated in Fig. 2. The upper right graph shows the resulting trade-off between $\text{PSNR}_e$ and $\Delta$PSNR and provides a compact overview of the overall system behavior. Because the curve shows the dependency between two distortion measures, we refer to it as the operational *Distortion – Distortion Function* (DDF). Note that the overall picture quality at the decoder, $\text{PSNR}_d$, increases from top-left towards bottom-right as illustrated in the figure. Therefore, if desired, DDFs can also be used to evaluate the overall distortion.

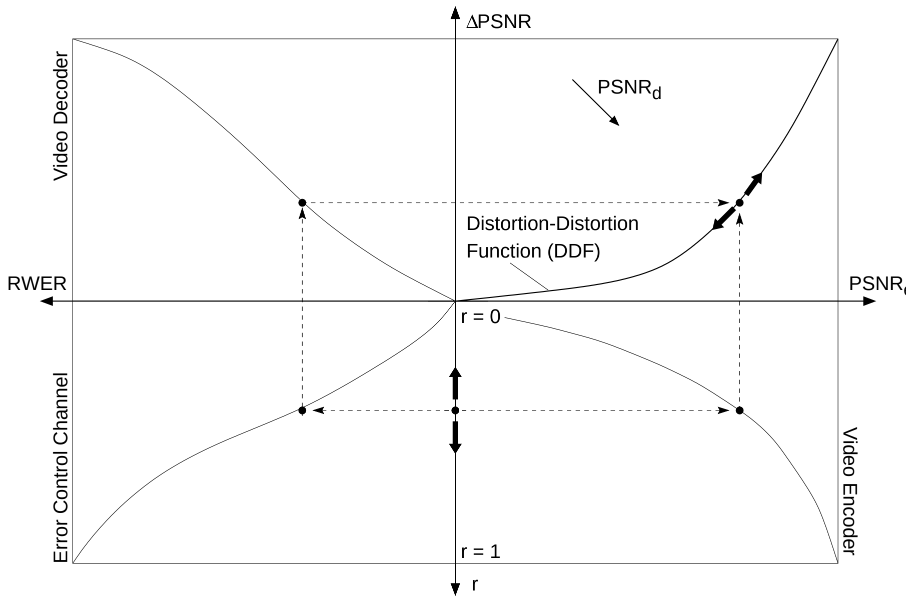


Figure 2: Interaction of system components when varying the bit allocation between source and channel coding, characterized by the channel code rate $r$. $\text{PSNR}_e$ is the picture quality after encoding and $\Delta$PSNR is the loss of picture quality caused by residual errors. An important system parameter of the error control channel is the residual word error rate (RWER). The upper right curve is the *Distortion – Distortion Function* of the wireless video system and is a compact description of the overall performance.

The DDF is a useful tool to study the influence of parameters or algorithms in the video codec for a given error control channel. Instead of building a combined distortion measure, both distortion types are available to evaluate the resulting system performance without additional assumption of how they have to be combined, as long as subjective quality decreases with both increasing $D_e$ and increasing $\Delta D$.

As pointed out in Section 2.2, $D_e$ is a useful distortion measure for source coding as long as the video signal is impaired by the same kind of distortion. More formally, let $Q$ be the average subjective video quality as perceived by a representative group of test persons. For $D_e$ to be useful for coder optimization, we require that $Q \approx f(D_e)$ for the set of impaired video sequences considered, where $f(.)$ is a monotonically decreasing function. The exact form of $f(.)$ is irrelevant. With a similar argument, $\Delta D$ is useful to optimize the error control channel and the video decoder, if the subjective quality $Q \approx g(\Delta D)$, where $g(.)$ is monotonically decreasing.

For the joint optimization of source and channel coding, we require a subjective quality function $Q \approx h(D_e, \Delta D)$ that captures the superposition of the two different types of distortions. Unfortunately, measuring $h(.,.)$ would require tedious subjective tests, and no such tests have been carried out to the authors' best knowledge. Nevertheless, we can safely assume that $h(.,.)$ would be monotonically decreasing with $D_e$ and $\Delta D$, and, fortunately, this monotonicity condition is often all we need when using DDFs to evaluate and compare error resilience techniques. In many situations, DDFs to be compared do not intersect in a wide range of $D_d$ and $\Delta D$. In this case it is possible to pick the best scheme for any $Q \approx h(D_e, \Delta D)$ as long as the monotonicity condition holds. This greatly simplifies the evaluation of video transmission systems, since the difficult question of a combined subjective quality measure for source coding and transmission error distortion is circumvented.

Fig. 3 illustrates two typical DDFs using PSNR$_e$ and $\Delta$PSNR as a quality measure. Because video codec B consistently suffers a smaller PSNR loss due to transmission errors, it is the better choice. Note that DDFs do *not* solve the problem of optimum bit allocation between source and channel coding, as this requires the knowledge of $h(.,.)$. In practical system design, the best bit allocation can be determined in a final subjective test, where different systems are presented that sample the best obtained DDF.

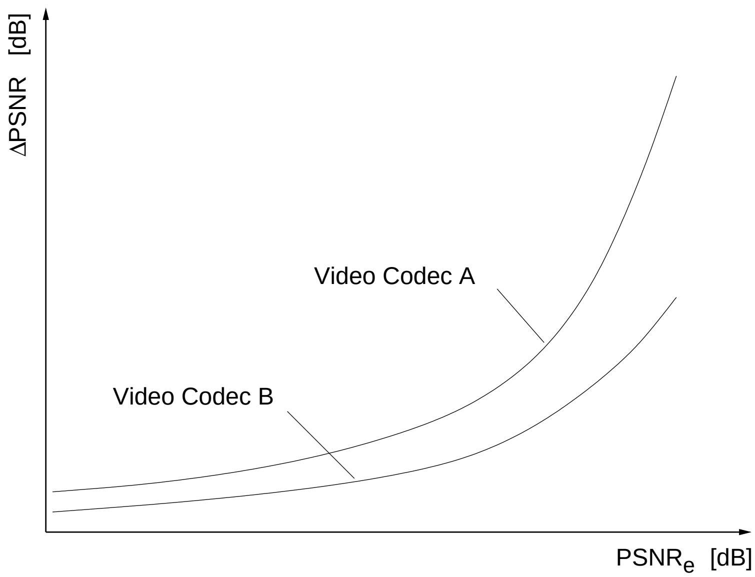


Figure 3: Distortion–Distortion Function (DDF) of two video codecs. Because codec B consistently provides a smaller PSNR loss ($\Delta$PSNR) for the same picture quality at the encoder (PSNR$_e$), it is the superior scheme.

## 3 Combating Transmission Errors

Before discussing error resilience techniques in the video encoder and decoder, we provide an introduction to error control techniques in the channel codec. Because the characteristics of the physical channel play an important role, we will first consider the properties of the mobile radio channel and the issue of modulation. For error control, we will focus on Reed-Solomon codes, interleaving, and automatic repeat request. Because of the increasing importance of open, Internet-style packet networks, we also consider the effect of packetization that can cause error amplification.

The following discussion also includes a description of the simulation environment that is used throughout this chapter. For modulation and channel coding we use standard components rather than advanced coding and modulation techniques. This is justified by our focus on video coding and by the fact that the selected standard components are well suited to illustrate the basic problems and trade-offs. Most of the conclusions that are derived in later sections also apply to other scenarios because the underlying concepts are very general. For more information on coding and modulation techniques that are employed in the next generation mobile networks, we refer to Chapter 5 "Wireless Networks". These are not discussed in detail below because the interface to the error control channel will behave similarly, resulting in similar problems and solutions on the source coding level.

## 3.1 Characteristics of the Mobile Radio Channel

The mobile radio channel is a hostile medium. Besides absorption, the propagation of electromagnetic waves is influenced by three basic mechanisms: reflection, diffraction, and scattering. In conjunction with mobility of the transmitter and/or receiver, these mechanisms cause several phenomena, such as time-varying delay spread or spectral broadening, which can severely impair the transmission. These will be briefly discussed in the following. More information can be found in Chapter 5 "Wireless Networks" or [28] [29] [30]. The intention of this section is to show that the underlying physical mechanisms result in fundamental performance limits for wireless transmission. As a result, the use of error control techniques in the video codec is of increased importance.

When a mobile terminal moves within a larger area, the distance between the radio transmitter and receiver often varies significantly. Furthermore, the number and type of objects between transmitter and receiver usually changes and might cause *shadowing*. The resulting attenuation of radio frequency (RF) power is described by the *path loss*. In an outdoor environment, the path loss is affected by hills, forests, or buildings. In an indoor environment, the electromagnetic properties of blocking walls and ceilings are of importance. The effect of these objects and the distance to the transmitter can be described by empirical models [28] [30]. Usually, these models include a mean path loss as a function of distance ($n$th power law) and a random variation about that mean (log-normal distribution). For our experimental results in this chapter we assume that the path loss is constant for the duration of a simulation (approximately 10 seconds), and hence assume constant (average) $E_b/N_0$.

Besides *large-scale fading* as described by path loss, small changes in position can also result in dramatic variations of RF energy. This *small-scale fading* is a characteristic effect in mobile radio communication and is caused by *multipath propagation*. In a wireless communication system, a signal can travel from transmitter to receiver over multiple reflective paths. Each reflection arrives from a different direction with a different delay and, hence, for a narrowband signal, undergoes a different attenuation and phase shift. The superposition of these individual signal components can cause constructive and destructive interference alternating at a small scale (as small as half a wavelength). For a moving receiver, this space-variant signal strength is perceived as a time-variant channel, where the velocity of the mobile terminal determines the speed of fluctuation. Small-scale fading is often associated with *Rayleigh fading* because, if the multiple reflective paths are large in number and equally significant, the envelope of the received signal is described by a Rayleigh pdf (probability density function).

An important problem caused by multipath propagation is *delay spread*. For a single transmitted impulse, the time $T_m$ between the first and last received components of significant amplitude represents the *maximum excess delay*, which is an important parameter to characterize the channel. If $T_m$ is bigger than the symbol duration $T_s$, neighboring symbols interfere with each other, causing *Intersymbol Interference* (ISI). This channel type requires special mitigation techniques such as equalization and will not be considered in the following. Instead, we focus on *flat fading* channels, with $T_m < T_s$. In this case, all the received multipath components of a symbol arrive within the symbol duration and no ISI is present. Here, the main degradation is the destructive superposition of phasor components, which can yield a substantial reduction in signal amplitude. Note, however, that the error resilience techniques described in Section 4 are also applicable to ISI channels given appropriate channel coding.

Similar to the delay spread in the time-domain, the received signal can also be spread in the frequency-domain. For a single transmitted sinusoid, the receiver may observe multiple signals at shifted frequency positions. This *spectral broadening* is caused by the Doppler shift of an electromagnetic wave when observed from a moving object. The amount of shift for each reflective path depends on the incident direction relative to the velocity vector of the receiver. The maximum shift magnitude is called the Doppler frequency $f_D$, which is equal to the mobile velocity divided by the carrier wavelength. For the dense-scatterer model, which assumes a uniform distribution of reflections from all directions, the resulting *Doppler power spectrum* has a typical bowl-shaped characteristic with maximum frequency $f_D$ (also known as Jakes spectrum [29]). This model is frequently used in the literature to simulate the mobile radio channel and is also used in this chapter. Note that the Doppler power spectrum has an important influence on the time-variant behavior of the channel because it is directly related to the temporal correlation of the received signal amplitude via the Fourier transform. For a given carrier frequency, the correlation increases with decreasing mobile velocity, such that slowly moving terminals encounter longer fades (and error bursts). Therefore, $f_D$ is often used to characterize how rapidly the fading amplitude changes.

In summary, mobile radio transmission has to cope with time-varying channel conditions of both large and small scale. These variations are mainly caused by the motion of the transmitter or the receiver resulting in propagation path changes. As a result, errors are not limited to single bit errors but tend to occur in bursts. In severe fading situations the loss of synchronization may even cause an intermittent loss of the connection. As we will see, this property makes it difficult to design error control techniques that provide high reliability at high throughput and low delay.

## 3.2 Modulation

Since we cannot feed bits to the antenna directly, an appropriate digital modulation scheme is needed. Usually, a sinusoidal carrier wave of frequency $f_c$ is modified in amplitude, phase, and/or frequency depending on the digital data that shall be transmitted. This results in three basic modulation techniques, known as Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK), and Phase Shift Keying (PSK). However, other schemes and hybrids are also popular. In general, the modem operates at a fixed symbol rate $R_s$, such that its output signal is cyclostationary with the symbol interval $T_s = 1/R_s$. In the most basic case, one symbol corresponds to one bit. For example, Binary PSK (BPSK) uses two waveforms with identical amplitude and frequency but a phase shift of 180 degrees. Higher order modulation schemes can choose from a larger set of waveforms, and hence provide higher bit-rates for the same symbol interval, but they are also less robust against noise for the same average transmission power.

The choice of a modulation scheme is a key issue in the design of a mobile communication system because each scheme possesses different performance characteristics. In most cases, however, the selection of a modulation scheme reduces to a consideration of the power and bandwidth availability in the intended application. For example, in cellular telephony the principle design goal is the minimization of spectral occupancy by a single user, such that the number of paying customers is maximized for the allocated radio spectrum. Thus, an issue of increasing importance for cellular systems is to select bandwidth-efficient modulation schemes. On the other hand, the lifetime of a portable battery also limits the energy that can be used for the transmission of each bit, $E_b$, and hence power efficiency is also of importance. A detailed discussion of modulation techniques is beyond the scope of this chapter and the reader is referred to [31] [32] for detailed information.

We conclude this section by describing the modulation scheme and parameters that are used for simulations in this chapter. Some of the modem parameters are motivated by the radio communication system DECT (Digital Enhanced Cordless Telecommunications). Though DECT is an ETSI standard originally intended for cordless telephony, it provides a wide range of services for cordless personal communications which makes it very attractive for mobile multimedia applications [33] [30]. Similar to DECT, we use BPSK for modulation and a carrier frequency of $f_c = 1900$ MHz. For moderate speeds (35 km/h) a typical Doppler frequency is $f_D = 62$ Hz, which will be used throughout the simulations in the remainder of this chapter. According to the double slot format of DECT, we assume a total bit rate of $R_c = 80$ kbps that is available for both source and channel coding. For simplicity we do not assume any TDMA structure and use a symbol interval of $T_s = 1/80$ ms. Note that the Doppler frequency $f_D$ together with $T_s$ determine the correlation of bit errors at the demodulator. Example bit error sequences shown in Fig. 4 exhibit severe burst errors.

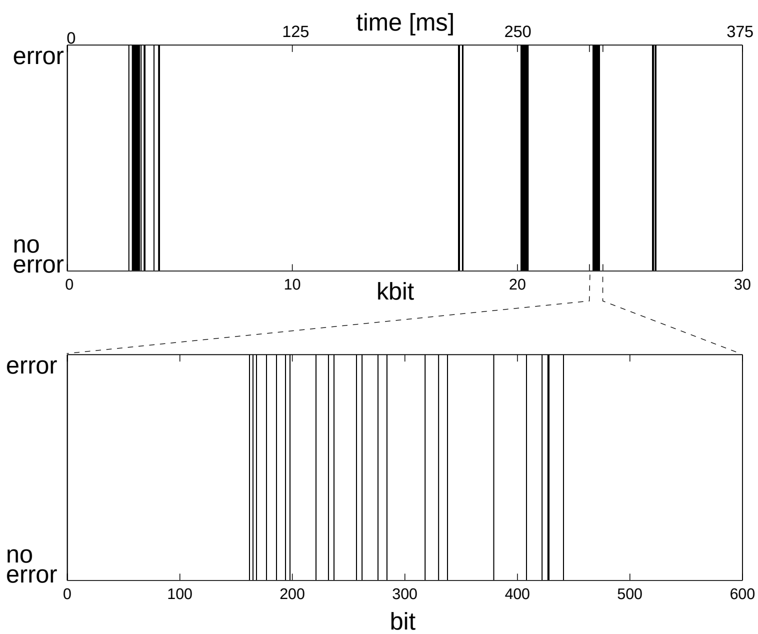


Figure 4: Illustration of burst errors encountered for Rayleigh fading channel (Doppler frequency $f_D = 62$ Hz, $E_b/N_0 = 18$ dB) and BPSK modulation (symbol interval $T_s = 1/80$ ms).

## 3.3 Channel Coding and Error Control

In this section, we discuss two main categories of channel coding and error control: FEC and ARQ. The latter requires a feedback channel to transmit retransmission requests, while FEC has no such requirement. We also address *Interleaving* as a way to enhance FEC in the presence of burst errors. In the following we will discuss the trade-off between (1) *throughput*, (2) *reliability*, and (3) *delay* of the error control channel and present some simulation results for illustration.

### Forward Error Correction

FEC techniques fall in two broad categories – block coding and convolutional coding. Though they are very different in detail, they both follow the same basic principle. At the transmitter, parity check information is inserted into the bitstream such that the receiver can detect and possibly correct errors that occur during transmission. The amount of redundancy is usually expressed in terms of the *channel code rate* $r$, which takes on values between zero (no payload information) and one (no redundancy). Though convolutional codes are as important in practice as block codes, we will use block codes to explain and illustrate the performance of FEC.

For block coding, the bitstream is grouped into blocks of $k$ bits. Then, redundancy is added by mapping $k$ information bits to a *code word* containing $n > k$ bits. Thus, the code rate of block codes is given by $r = k/n$. The set of $2^k$ code words is called the *channel code* $C(n,k)$. For a *systematic code*, the $k$ information bits are not altered and $n-k$ check bits are simply appended to the payload bits. Decoding is achieved by determining the most likely transmitted code word given a received block of $n$ bits.

The error correction capability of a $C(n,k)$ code is primarily influenced by the *minimum Hamming distance* $d_{min}$. The Hamming distance of two binary code words is the number of bits in which they differ. For a code with minimum Hamming distance $d_{min}$ the number of bit errors that can be corrected is at least

$$t = \lfloor (d_{min} - 1)/2 \rfloor.$$

Therefore, the design of codes, i.e., the selection of $2^k$ code words from the set of $2^n$ possible codewords, is an important issue as $d_{min}$ should be as large as possible. For large $n$, this is not straightforward, especially when also considering the problem of decoding. Furthermore, there are fundamental limits in the maximization of $d_{min}$, such as the *Singleton bound*

$$d_{min} \leq n - k + 1.$$

Fortunately, channel coding is a mature discipline that has come up with many elegant and clever methods for the nontrivial tasks of code design and decoding algorithms. In the following, we will limit the discussion to the *Reed-Solomon* (RS) codes as a particularly useful class of block codes that actually achieve the Singleton bound. Other block codes of practical importance include Bose-Chaudhuri-Hocquenghem (BCH), Reed-Muller, and Golay codes [23].

Reed-Solomon codes are used in many applications, ranging from the compact disk (CD) to mobile radio communications (e.g. DECT). Their popularity is due to their flexibility and excellent error correction capabilities. RS codes are non-binary block codes that operate on multi-bit symbols rather than individual bits. If a symbol is composed of $m$ bits, the RS encoder for an $RS(N,K)$ code groups the incoming data stream into blocks of $K$ information symbols ($Km$ bits) and appends $N-K$ parity symbols to each block. For RS codes operating on $m$-bit symbols, the maximum block length is $N_{max} = 2^m - 1$. By using *shortened* RS codes, any smaller value for $N$ can be selected, which provides a great flexibility in system design. Additionally, $K$ can be chosen flexibly, allowing a wide range of code rates. Later on, we will take advantage of this flexibility to investigate different bit allocations between source and channel coding.

Let us now consider the error correction capability of an $RS(N,K)$ code. Let $E$ be the number of corrupted symbols in a block containing $N$ symbols. Note that a symbol is corrupted when any of its $m$ bits is in error. Though this seems to be a drawback for single bit errors, it can actually be advantageous for the correction of burst errors, as typically encountered in the mobile radio channel. As RS codes achieve the Singleton bound, the minimum number of correctable errors is given by

$$T = \lfloor (N-K)/2 \rfloor,$$

and the RS decoder can correct any pattern of symbol errors as long as $E \leq T$. In other words, for every two additional parity symbols, an additional symbol error can be corrected. If more than $E$ symbol errors are contained in a block, the RS decoder can usually detect the error. For large blocks, undetected errors are very improbable, especially when the decoder backs off from the Singleton bound for error correction (*bounded distance decoding*). The probability that a block cannot be corrected is usually described by the *Residual Word Error Rate* (RWER). In general, the RWER decreases with increasing $K$ and with increasing $E_b/N_0$.

The performance of RS codes for the mobile radio channel discussed previously is shown in Fig. 5. On the left side we show the RWER for a variation of $r$. For a given value of $E_b/N_0$, the RWER can be reduced by approximately one to two orders of magnitude by varying the code rate in the illustrated range. This gain in RWER is very moderate due to the bursty nature of the wireless channel. For channels without memory, such as the additive white Gaussian noise (AWGN) channel, the same reduction in $r$ would provide a significantly higher reduction in RWER. In this case it is possible to achieve very high reliability (RWER $< 10^{-6}$) with only little parity-check information and resilience techniques in the video codec would hardly be necessary. For bursty channels, however, the effectiveness of FEC is limited as the error correction capability is often exceeded when a block is affected by a burst. Note that the left side of Fig. 5 illustrates the trade-off between *throughput* ($r$) and *reliability* (RWER) of the error control channel.

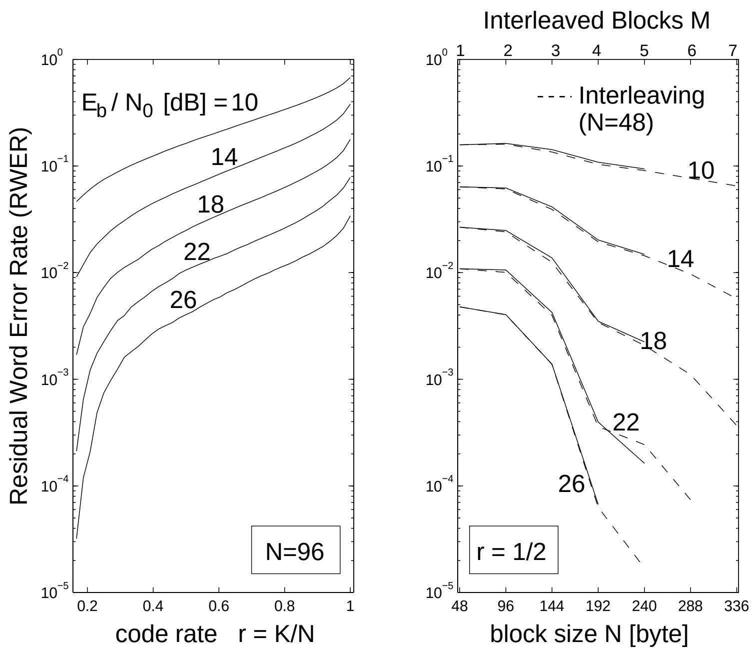


Figure 5: Residual word error rate (RWER) for the variation of channel code rate $r$ (left) and block size $N$ (right). Rayleigh fading with BPSK modulation and Reed-Solomon codes operating on 8-bit symbols are assumed.

The right side of Fig. 5 shows the RWER for a variation of the block length $N$. The dashed lines will be considered later. From the solid lines it can be seen that the increase in block length can be very effective for high $E_b/N_0$. Note that the throughput is not affected, as the code rate is kept constant at $r = 1/2$. However, the trade-off between *reliability* (RWER) and *delay* ($N$) has to be considered when choosing $N$. On the one hand, the error correction capability of a block code increases with the block length. On the other hand, long blocks introduce additional delay (assuming constant code rate and source rate). Usually the acceptable delay is determined by the application. For file transfer high delays in the order of several seconds are acceptable. For conversational services, such as voice or video telephony, a maximum round trip delay of 250 ms should not be exceeded. For low-delay video applications, the frame interval usually sets the upper bound. For example, assuming 12.5 fps video and a total bit-rate of $R_c = 80$ kbps, the resulting maximum block length is $n = 6400$ bit. However, shorter blocks are preferable because other effects also contribute to the overall delay.

Besides the limitations on $N$ which are imposed by delay considerations, there are also implementation and complexity constraints. In particular the *decoding* of block codes in case of errors is a task that becomes computationally demanding for large $N$. The number of bits that are combined to symbols in RS codes is usually less than and most commonly equal to 8, thus allowing a maximum block length of $N_{max} = 2^m - 1 = 255$ bytes.

Note that a limited block length can cause severe problems for FEC schemes when the transmission channel tends to burst errors. Either a block is affected by a burst, in which case the error correction capability is often exceeded, or the block is transmitted error-free and the additional redundancy is wasted. To overcome this limitation, FEC is often enhanced by a technique known as *Interleaving*.

### Interleaving

The idea behind interleaving is to spread the error burst in time. In a simple block interleaver, encoded blocks of $N$ symbols are loaded into a rectangular matrix row by row. After $M$ rows are collected, the symbols are then read out column by column for transmission. At the receiver, this reordering of symbols is inverted and the blocks are passed to the FEC decoder. For burst errors, this effectively reduces the concentration of errors in single code words, i.e., a burst of $b$ consecutive symbol errors causes a maximum of $b/M$ symbol errors in each code word. For large $M$, the interleaver/-deinterleaver pair thus creates in effect a memoryless channel. Though interleaving can be implemented with low complexity it also suffers from increased delay, depending on the number of interleaved blocks $M$. The dashed lines on the right side of Fig. 5 illustrate the effectiveness of interleaving in the given error control channel. As the basic block length we use $N = 48$ symbols. For the same delay, essentially the same performance can be achieved as for increased block length, providing the same trade-off between *reliability* and *delay*. However, also larger blocks than $N_{max} = 255$ can be obtained at reduced complexity. Therefore interleaving is a frequently used technique for bursty channels if the additional delay is acceptable.

### Automatic Repeat Request

Another error control technique that can be used to exchange reliability for delay and throughput is *Automatic Repeat reQuest* (ARQ). In contrast to FEC, ARQ requires a feedback channel from the receiver to the transmitter, and therefore cannot be used in systems where such a channel is not available (e.g., broadcasting).

For ARQ, the incoming bitstream is grouped into blocks, similar to FEC. Each block is extended by a header including a *Sequence Number* (SN) and an error detection code at the end of each block – often a *Cyclic Redundancy Check* (CRC). This information is used at the receiver for error detection and to request the retransmission of corrupted blocks using *Positive Acknowledgments* (ACKs) and/or *Negative Acknowledgments* (NAKs) which are sent back via the feedback channel. Usually, retransmissions are repeated until error-free data are received or a time-out is exceeded.

This basic operation can be implemented in various forms with different implications on throughput, complexity and delay. There are three basic ARQ schemes in use: *Stop And Wait* (SW), *Go Back N* (GN), and *Selective Repeat* (SR) [6]. Though SR-ARQ requires buffering and reordering of out-of-sequence blocks, it provides the highest throughput. Another possibility to enhance ARQ schemes is the combination with FEC, which is known as *Hybrid ARQ*. For a detailed analysis of throughput the reader is referred to [23] and [34], both of which also consider the case of noisy feedback channels. Furthermore, the application of ARQ in fading channels is analyzed in [35], while [36] proposes an ARQ protocol that adapts to a variable error rate by switching between two modes.

One critical issue in ARQ is delay, because the duration between retransmission attempts is determined by the *Round Trip Delay* (RTD). Thus, if the number of necessary retransmission attempts is $A$, the total delay until reception is at least $D = A \cdot \mathrm{RTD}$. As $A$ depends on the quality of the channel, the resulting delay and throughput are not predictable and vary over time. For applications, where delay is not critical, ARQ is an elegant and efficient error control technique, and it has been used extensively, e.g., in the *Transport Control Protocol* (TCP) of the Internet. For real-time video transmission, however, the delay associated with classic ARQ techniques is often unacceptable.

The situation has improved slightly in the past few years through delay-constrained or *soft* ARQ protocols. One simple approach to limit delay with ARQ is to allow at most $A = D/\mathrm{RTD}$ retransmissions, where $D$ is the maximum acceptable delay. As this may result in residual errors, the trade-off *reliability* vs. *delay* has to be considered. A given maximum-delay constraint can also be met by adjusting the source rate of the video codec. If a close interaction between source coding and channel is possible, the rate of the video codec can be directly controlled by the currently available throughput [37] [38]. The effectiveness of this approach for wireless video transmission over DECT has been demonstrated already in 1992 [7]. If such a close interaction is not possible, scalable video coding has to be used [8] [39] [40]. Other refinements of ARQ schemes proposed for video include the retransmission of more strongly compressed video [41] or the retransmission of multiple copies [9] in a packet network. Nevertheless, ARQ can only be used for applications with relatively large acceptable delay and/or very low RTDs – or with limited reliability.

## 3.4 IP over Wireless

Future wireless video applications will have to work over an open, layered, Internet-style network with a wired backbone and wireless extensions. Therefore, common protocols will have to be used for the transmission across the wired and the wireless portion of the network. These protocols will most likely be future refinements and extensions of today's protocols built around the *Internet Protocol* (IP). One issue that arises when operating IP over a wireless radio link is that of *fragmentation and reassembly* of IP packets. Because wireless radio networks typically use frame sizes that are a lot smaller than the maximum IP packet size, big IP packets have to be fragmented into several smaller packets for transmission and reassembled again at the receiving network node. Unfortunately, if any one of the small packets is corrupted, then the original big packet will be dropped completely, thus increasing the effective packet loss rate. One way to avoid fragmentation is to use the minimum packet size along the path from the transmitter to the receiver. However, this information is usually not available at the terminal. Furthermore, the overhead of the IP packet headers (typically 48 bytes with IP/UDP/RTP) may become prohibitive.

The resulting error amplification is illustrated in Fig. 6 for the investigated error control channel, where the fragments of the IP packet are mapped to FEC blocks. As can be seen, the effective RWER increases with the number of blocks that are necessary to convey the original packet. This problem has to be considered for the design of future multimedia-aware protocols, e.g., by allowing corrupt fragments within the reassembled IP-packet which are indicated in the header. However, we will not further discuss this (and other) problems that arise from intermediate network layers. Here, our intention is merely to point out that there are important protocol enhancements to be considered to avoid unnecessary performance degradation.

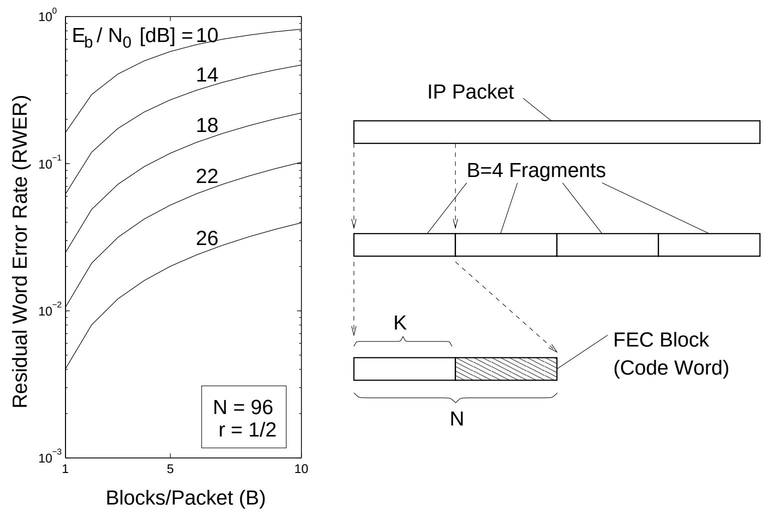


Figure 6: Residual word error rate (RWER) after fragmentation and reassembly of a packet into $B$ blocks. Rayleigh fading with BPSK modulation and Reed-Solomon codes operating on 8-bit symbols are assumed.

# 4 Error Resilience Techniques for Low Bit-Rate Video

In this section we focus on error resilience techniques that can be used in the video codec to improve the overall performance. As an example, we use the important H.263 video compression standard. Most of the ideas discussed can also be applied to other video coding algorithms, and are also covered in Chapter 8, “Error Resilience Coding”. We assume that the reader is familiar with the basic concepts and terms of the H.263 video compression standard. For a review, we refer to Part I, “Compressed Video Standards” or [42] [43].

Although we restrict our investigation to methods that are supported by the current H.263 syntax, it is still very difficult to provide a complete analysis. On the one hand, H.263 includes various optional coding modes that can be used for error resilience in many combinations. On the other hand, the operation of the encoder is not standardized, such that the present syntax can be used in a very flexible way. The clever combination of existing options that may not even be intended for error resilience can significantly increase the performance of a wireless video system. Furthermore, the operation of the decoder in the case of errors is not covered by the current version of the H.263 standard. Although ITU-T Study Group 16 intends to include error detection and error concealment in future versions of Recommendation H.263, it is not covered at present and the performance of the decoder therefore depends heavily on a particular implementation. Because of this flexibility in mode combinations and decoder operation, we can only discuss the most common and effective error resilience techniques.

## 4.1 Input Format and Rate Control

To achieve high compression, as required for the transmission over mobile radio channels at low bit-rates, both the spatial resolution and the frame rate have to be reduced compared to standard television pictures. We use the QCIF format (Quarter Common Intermediate Format, $176 \times 144$ pixels) for our simulations, which is the most common input format at the considered range of bit-rates. As a typical frame rate we use 12.5 fps (frames per second). Though a variable frame rate in the range of 5 to 15 fps may be advantageous for subjective quality, we maintain a fixed rate of 12.5 fps to allow a fair comparison between different approaches based on PSNR values. Unless otherwise stated, we use sequences of 300 frames, i.e., 150 encoded frames covering a time period of 12 seconds. Because the transmission over an unreliable channel introduces random errors, several simulations have to be performed for different channel realizations to obtain averaged results according to (2). For each investigated error resilience technique and parameter setting (FEC, $E_b/N_0$, INTRA percentage, ...) we use $L = 30$ channel realizations.

We use a simple rate control in our simulations. Each frame is encoded with a fixed quantizer step size, which is adapted frame by frame to obtain a given target bit rate. The adaptation of the quantizer step size is performed as follows. First the mode decision is performed according to TMN5 [44] for the whole frame, and then the resulting prediction error is transformed and quantized with different quantizer step sizes. Finally, the value that minimizes the deviation from a desired buffer fullness is selected. This rate control reduces buffer variations to an acceptable amount, and hence allows the transmission over a constant bit-rate channel with limited delay. In practice, more sophisticated quantizer control algorithms could be used that can further reduce buffer fluctuations at improved rate-distortion performance. For more information on rate control algorithms and their implications on performance and delay, see Chapter 9, "Variable Bit-Rate Video Coding."

## 4.2 Error Detection and Resynchronization

Transmission errors can be detected in a variety of ways. With FEC, errors can often be detected with high reliability by the channel decoder, even if the correction capability of the code is exceeded. For example, in H.261 and H.263 an optional FEC framing can be used to detect errors within a BCH(511,493) code word. If a packet protocol stack such as IP is used, lower layers often provide error detection as a basic service. For example, this is the case for the packet video standard ITU-T Rec. H.323 (see Chapter 13, "Networked Video Systems Standards"). For our simulations, an RS code is used with a block size of 88 bytes. This block size corresponds to the average size of one GOB for the given input format (QCIF, 12.5 fps) and bit-rate (80 kbps). The packetization delay in the order of a GOB can be neglected, even for low-latency applications.

In some transmission systems, reliability information can be obtained for each received bit when the receiver provides channel state information, or when the channel decoder provides reliability information [45] [46]. This information is then passed on to the video decoder. In addition, the video decoder itself can detect transmission errors. The video bit-stream contains some residual redundancy, such that violations of syntactic or semantic constraints will usually occur quickly after a loss of synchronization [13] [47] [48], [49]. For example, the decoder might not find a matching variable length code (VLC) word in the code table (a syntax violation), or detect that the decoded motion vectors, DCT coefficients, or quantizer step-sizes exceed their permissible range (semantic violations). Additionally, the accumulated run that is used to place DCT coefficients into an $8 \times 8$ block might exceed 64, or the number of MBs in a GOB might be too small or too large. Especially for severe errors, the detection can further be supported by localizing visual artifacts that are unlikely to appear in natural video signals. However, these inherent error detection capabilities of a video decoder have a limited reliability and cannot exactly localize errors. Usually, several erroneous code words are processed by the video decoder before syntactic or semantic constraints are violated, and the distance between error location and error detection can vary significantly. Therefore, external error detection mechanisms, such as FEC framing, should be used if available.

A more difficult problem than error detection is resynchronization after a detected error. Because the multiplexed video bit-stream consists of VLC words, a single bit error often causes a loss of synchronization and a series of erroneous code words at the decoder. Residual redundancy in non-compact VLCs can be used to design self-synchronizing codes, such that valid symbols may be obtained again after some slippage [50]. However, even if resynchronization is regained quickly, the appropriate location of the decoded information within the video frame is no longer known, since the number of missing symbols cannot be determined. Moreover, the subsequent code words are useless if the information is encoded differentially, as it is often the case, e.g., for motion vectors. The common solution to this problem is to insert unique synchronization code words into the bit-stream in regular intervals, usually followed by a block of "header" bits. Since any conditional encoding across the resynchronization point must be avoided, the header provides anchor values, e.g., for absolute location in the image or current quantizer step size. Although the length of the synchronization code word can be minimized [51], relatively long synchronization code words are used in practice to reduce the probability of accidental emulation in the bitstream.

Recently, more advanced techniques for resynchronization have been developed in the context of MPEG-4 and H.263++. Among several error resilience tools, *data partitioning* has been shown to be effective [13]. Especially when combined with *reversible variable length coding* (RVLC), which allows bit-streams to be decoded in either forward or reverse direction, the number of symbols that have to be discarded can be reduced significantly. Because RVLCs can be matched well to the statistics of image and video data, only a small penalty in coding efficiency is incurred [52] [53]. A recently proposed technique can even approach the efficiency of Huffman codes by combining a prefix and suffix code word stream by delayed XORing [54]. Another elegant technique that is not part of any current video coding standard has been proposed by Redmill and Kingsbury as *Error-Resilient Entropy Coding* (EREC) [55]. Similar to data partitioning, a re-ordering of the bit-stream is involved. Instead of clustering all symbols of the same type into one partition, EREC re-organizes VLC image blocks such that each block starts at a known position within the bit-stream and the most important information is closest to these synchronization points. More information on RVLCs and EREC can also be found in Chapter 8, "Error Resilience Coding".

Considering H.263, the most basic way of improving resynchronization is to use more GOB-headers, which can be inserted optionally as resynchronization points at the beginning of each GOB. In QCIF format, 9 GOBs are encoded which consist of 11 macroblocks in one row. Except for the first GOB that always starts with the picture-header all GOBs may or may not include a GOB-header. The unique synchronization word that is used in H.263 as a preamble for the GOB-header consists of 16 consecutive 0-bits followed by a 1-bit ("00000000000000001"). The encoding of anchor values in the header that follows the synch word require 12 bits, such that the total number of bits for a GOB-header is 29. In addition to this overhead, the rate-distortion performance is further reduced by less effective prediction of motion vectors. However, this reduced coding efficiency is usually well compensated by improved error resilience as will be shown below.

Because all information between two resynchronization points can be used independently from previous information in the same frame, the corresponding set of macroblocks is often used as the basic unit for decoding. In the following we use the term *slice* to refer to this set of macroblocks. In the baseline mode of H.263, a slice always corresponds to an integer number of GOBs because the placement of resynchronization points is restricted to the start of a GOB. However, in the *slice structure mode* of H.263 (Annex K) the placement of resynchronization points is allowed after each macroblock providing increased flexibility. On the one hand, it is possible to further increase the number of resynchronization points per frame, which is restricted to 9 in QCIF baseline mode. On the other hand, it is possible to adapt the size of slices to the size of packets or FEC blocks. When Annex K is enabled, GOB-headers are replaced by Slice-headers with very similar functionality but slightly increased overhead (34 bits including synch word).

Typically, if a transmission error is detected within a slice, it is discarded entirely and error concealment is invoked for all its macroblocks. This approach is also taken throughout this chapter for the experiments. Because we employ an FEC framing with fixed block size, several slices may overlap with a single FEC block. In this case, the decoder invokes error concealment for each slice that overlaps with the corrupted block. To reduce the number of discarded macroblocks it is therefore particularly important to reduce the size of slices. In the *slice structure mode*, the overlap of an FEC blocks with two adjacent slices can be avoided by starting a new slice at the beginning of the next block. This results in fixed length "video packets" that include a variable number of macroblocks. In MPEG-4, this technique has already proven to be effective. However, this ideal packetization requires a close interworking between source and channel coding, which may be difficult in some situations. For example, in a call through a gateway, the source coding is done in a terminal, while the packetization is done in the gateway. Fig. 7 schematically illustrates the advantage of using an increased number of slices per frame and the additional advantage of alignment with packets.

Finally, Fig. 8 illustrates the performance of H.263 using different numbers of synch words per frame. The *slice structure mode* is not enabled, but the number of synch words is increased by inserting additional GOB-headers. Discarded macroblocks are concealed by simply copying the corresponding image data from the reference frame (see Section 4.3). From the left side of Fig. 8 it can be seen that the maximum number of headers (one picture header and 8 GOB headers) is always advantageous for $E_b/N_0$ = 26 dB. Though the loss in picture quality $\Delta$PSNR is only slightly improved by using more than 5 synch words per frame, the increased overhead for 9 synch words is still justified. However, it can be expected that a significantly higher number, as would be possible using Annex K, would finally result in a reduced performance. In Chapter 8, "Error Resilient Coding", the optimum amount and location of synch words is investigated for the slice structure mode. The right side of Fig. 8 that the loss in picture quality when a PSNR of 36 dB is required at the encoder output. For the whole range of investigated $E_b/N_0$ values, 9 synch words/frame provide the optimum performance. For all other simulations in this chapter we will therefore use GOB-headers for each GOB.

## 4.3 Error Concealment

The severeness of residual errors can be reduced if error concealment techniques are employed to hide visible distortion as well as possible. Since typically an entire GOB is affected (i.e., 16 successive luminance lines), spatial interpolation is less efficient than

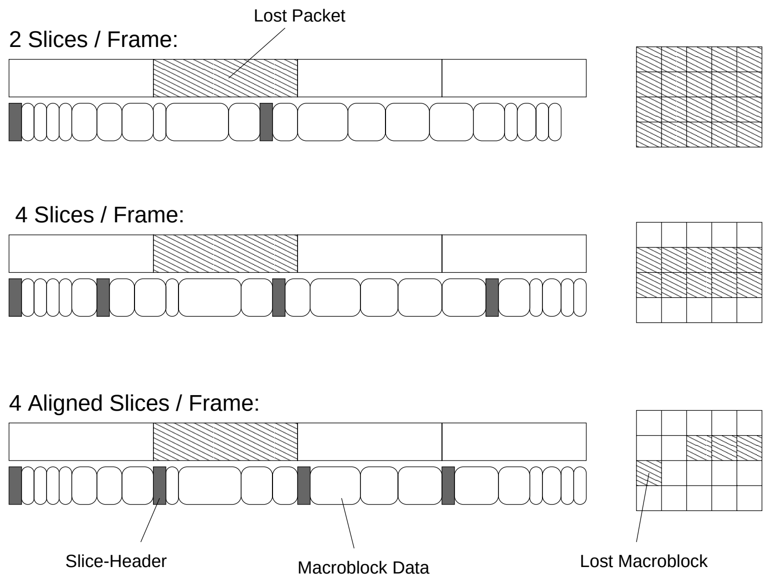


Figure 7: Illustration of resynchronization techniques that can be used in H.263. On the left, the variable length macroblocks in the bitstream are illustrated together with the fixed size packets that are used for transmission. On the right, the discarded macroblocks are illustrated in the reconstructed frame. For the illustration, a reduced frame size of 5×4 macroblocks is used instead of the 11×9 macroblocks in the QCIF format.

temporal extrapolation. Only in the case of very complex motion or scene cuts, it can be advantageous to rely on the spatial correlation in the image [56] [57], or switch between temporal and spatial concealment [58] [49]. In the simplest and most common approach, *previous frame concealment*, the corrupted image content is replaced by corresponding pixels from the previous frame. This simple approach yields good results for sequences with little motion [47]. However, severe distortions are introduced for image regions containing heavy motion.

If data partitioning and strong error protection for motion vectors is used, one might rely on the transmitted motion vectors for motion-compensated concealment. If motion vectors are lost, they can be reconstructed by appropriate techniques, for example, by spatial interpolation of the motion vector field [59], which can be enhanced by additionally considering the smoothness of the concealed macroblock along the block boundaries [60] [61]. The interested reader is referred to Chapter 6, "Error Concealment," or [9] for a comprehensive overview of concealment techniques. All error resilience techniques discussed in the sequel benefit similarly from better concealment. Hence, it suffices to select one technique, and we present experimental results for the simple *previous frame concealment* in the following sections. Only in this section, we also investigate a second concealment technique that we refer to as *encoded motion concealment*. In this approach, the motion vectors that are used at the encoder for prediction are also used at the decoder for motion-compensated concealment. Though this approach cannot be implemented in practice, as it assumes error-free transmission of motion vectors, it provides an upper bound for the performance of motion-compensated concealment. Therefore it becomes possible to relate the gains obtained by error concealment to the gains obtained by other error resilience techniques. However, as pointed out above, error concealment is always an enhancement of, not a replacement for, other error resilience techniques.

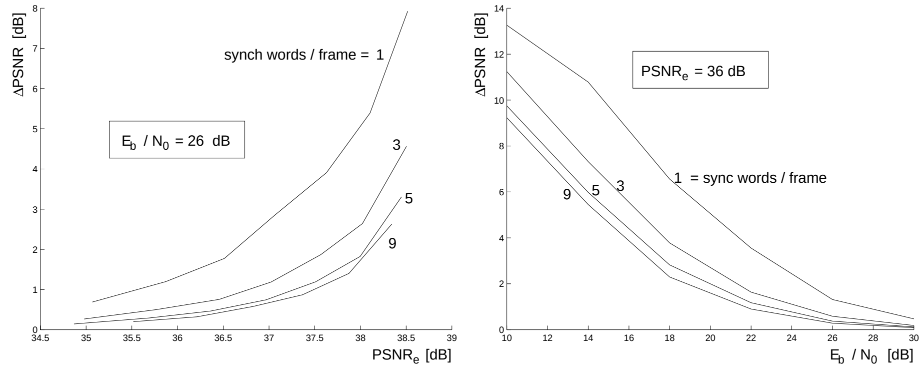


Figure 8: Evaluation of video quality using different numbers of synch words per frame. Left: operational DDF for fixed channel. Right: $\Delta$PSNR for fixed PSNR$_e$. The test sequence is *Mother and Daughter*.

Fig. 9 (left) illustrates the possible gain of encoded motion concealment over previous frame concealment for $E_b/N_0 = 22$ dB. The improvement in $\Delta$PSNR becomes most obvious for high residual error rates, i.e., when less bits are assigned to the channel codec and the video encoder can obtain high values for PSNR$_e$. At PSNR$_e$ = 36 dB the loss in picture quality is reduced from 1 to 0.5 dB. However, it should be noted that the test sequence used contains only moderate motion. For sequences with more complex motion, the difference is more severe. From the right side of Fig. 9 it can be seen that the gain obtained by error concealment can be converted into improved power efficiency for the transmission. For a given loss in picture quality, e.g., $\Delta$PSNR = 1 dB, the required value of $E_b/N_0$ can be reduced by approximately 4 dB. In practice, however, this reduction in transmission power may have to be paid for by increased computation for error concealment. Nevertheless, concealment is an important error resilience technique, because it provides significant gains at no overhead in bit-rate.

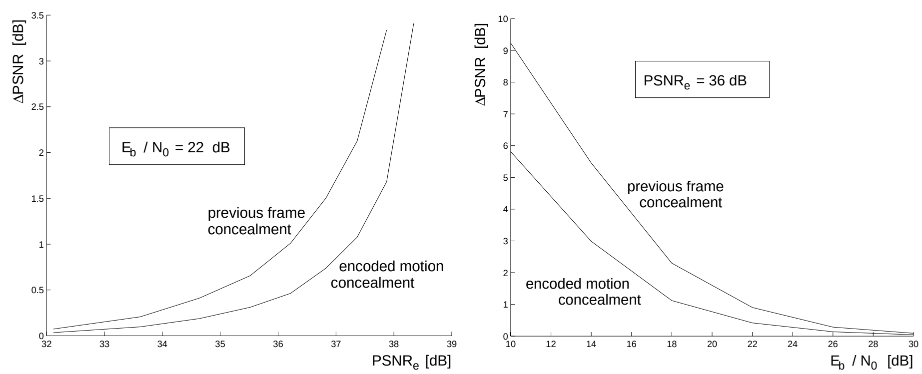


Figure 9: Evaluation of video quality using different error concealment techniques. Left: operational DDF for fixed channel. Right: $\Delta$PSNR for fixed PSNR$_e$. The test sequence is *Mother and Daughter*.

## 4.4 Mitigation of Interframe Error Propagation

Resynchronization as well as error concealment reduce the amount of image distortion that is introduced for a given error event, e.g., a packet loss. However, they do not prevent error propagation which is the most dominant and annoying error effect in video coding. Error propagation is caused by the recursive structure of the decoder when operating in the interframe mode. In this mode, the previously decoded frame is used as a reference for the prediction of the current frame. Errors remaining after concealment therefore propagate to successive frames and remain visible for a long period of time. In addition, the accumulation of several errors can result in very poor image quality, even if the individual errors are small. Fig. 10 illustrates the typical transmission error effects for the loss of one GOB in frame 4. Not only does the error propagate temporally, but it also spreads spatially due to motion-compensated prediction. In the remaining part of this chapter, we focus on error resilience techniques that mitigate the effect of error propagation in various ways. In general, three basic approaches are possible to remove errors from the prediction loop, once they are introduced.

- The prediction from previous frames is omitted by using the INTRA mode.
- The prediction from previous frames is restricted to error free image regions.
- The prediction signal is attenuated by *leaky prediction.*

From a theoretical point of view, the first and last items are related, since INTRA coding can be considered as an extreme form of leaky prediction, where the prediction signal is completely attenuated. By using the INTRA mode for a certain percentage of the coded sequence, it is also possible to adjust the average attenuation. However, leaky prediction is a more general scheme that provides additional flexibility. Furthermore, leaky prediction is not explicitly supported by existing standards for improved error resilience. We therefore discuss both items separately in the following.

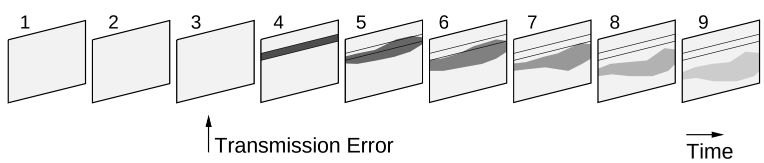


Figure 10: Illustration of spatio-temporal error propagation.

#### 4.4.1 Leaky Prediction

Leaky prediction is a well-known technique to increase the robustness of Differential Pulse Code Modulation (DPCM) systems by attenuating the energy of the prediction signal [62]. Because the attenuation is applied in each time step, the energy of superimposed errors decays over time and is finally reduced to a negligible amount. In contrast to speech and still image coding, this technique has not received a lot of attention in recent contributions to error resilient video coding, even though the idea is not new [63] [59]. Nevertheless, the underlying effect plays an important role in interframe error propagation of current video codecs, because leakage is introduced as a side-effect by spatial filtering in the motion-compensated predictor.

H.263 and all recent video compression standards employ bilinear interpolation for sub-pixel motion compensation, which acts as a lowpass filter. Spatial filtering in the motion-compensated predictor is a necessary ingredient for good compression performance of a hybrid coder [64] [65]. Even with integer-pixel accurate motion compensation, a “loop filter” should be employed. For example, in H.261, which uses integer-pixel motion compensation, the PSNR gain due to the loop filter is up to 2 dB [2] [43]. As lowpass filtering attenuates the high spatial frequency components of the prediction signal, leakage is introduced in the prediction loop. While error recovery is also improved at the same time, this is really a side-effect, and the leakage in the DPCM loop of standardized video codecs by itself is not strong enough for error robustness. For this purpose, additional leakage, such as more severe lowpass filtering could be introduced. Although this would reduce coding efficiency, the trade-off between coding efficiency and error resilience may be more advantageous than for INTRA coding because of increased flexibility in the design of the loop filter.

Considering the standardized H.263 syntax, the possible influence on the spatial loop filter and the leakage in the prediction loop is limited, especially when operating in the baseline mode. Though it is possible to prefer half-pel motion vectors over integer-pel motion vectors, the obtained gain is usually small. However, several coding modes of H.263 add additional spatial filtering to the motion-compensated predictor. For example, the *advanced prediction mode* (Annex F) uses overlapped block motion compensation (OBMC) and the *deblocking filter mode* (Annex J) introduces additional filtering at the borders of coarsely quantized blocks. As these options also improve coding efficiency, they can have a two-fold advantage for a wireless video transmission system.

Also in the baseline mode, the influences of leakage on interframe error propagation in H.263 cannot be neglected. This is shown in Fig. 11 that illustrates the recovery of $\Delta$PSNR after the loss of one GOB when previous frame concealment is used. The QCIF sequence *Foreman* is coded in the baseline mode at 100 kbps and 12.5 fps, resulting in an average $\mathrm{PSNR}_e$ of about 34 dB in the error-free case. Using the reconstructed frames at the encoder as the baseline, the loss in picture quality is calculated for each reconstructed frame at the decoder output. The figure shows the average degradation when each individual GOB in the fifth encoded frame is lost, one at a time. Though the error energy decays over time, a residual loss of 1 dB still remains in the sequence after 3 seconds. Note that no macroblocks are encoded in INTRA mode, and therefore the decay is entirely caused by spatial filtering. In contrast to leaky prediction in one-dimensional DPCM systems, where the error energy decays exponentially, it can be shown that the decay in MSE is roughly proportional to $1/t$ in hybrid video coding [66] [67]. More precisely, the MSE at the decoder, after a transmission error that introduces the additional error $\mathrm{MSE}_0$, can be modeled by

$$\mathrm{MSE}_d[t] = \mathrm{MSE}_e + \frac{\mathrm{MSE}_0}{1+\gamma t},$$

where $\gamma$ is a parameter that describes the effectiveness of the loop filter to remove the introduced error, typically in the range $0 < \gamma < 1$, and $\mathrm{MSE}_e$ is the MSE at the encoder. $t$ is the frame index, with the transmission error occurring at $t = 0$. As can be seen from Fig. 11, this model can describe experimental results quite accurately when the parameters are chosen appropriately.

Because the amount of leakage in H.263 is too small to be useful for error resilience, other techniques are needed to limit interframe error propagation. The most common approach, which can be implemented easily in H.263, is the regular INTRA update of image regions.

### 4.4.2 INTRA Update

A reliable method to stop interframe error propagation is the regular insertion of I-frames, i.e., pictures that are encoded entirely in INTRA mode. Unfortunately, for the same picture quality I-frames typically require several times more bits than frames encoded with reference to previous frames. Therefore, their use should be restricted as much as possible, especially for low bit-rates. Because the regular insertion of I-frames causes a significant variation in bit-rate, which has to be smoothed by buffering for constant bit-rate transmission, additional delay in the order of several frame intervals

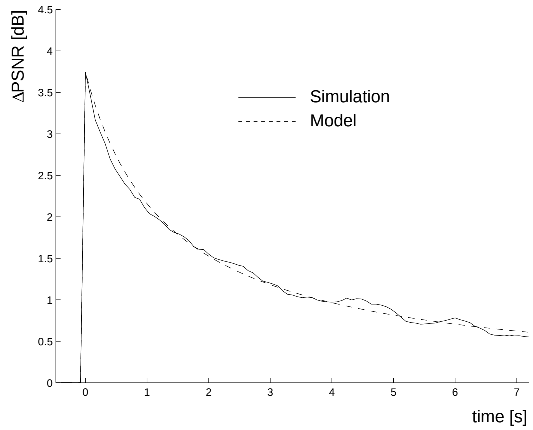


Figure 11: Recovery of picture quality after previous frame concealment of one GOB. Leakage introduced by spatial filtering is the cause of this recovery.

is introduced. To avoid this, INTRA coding can be distributed over several frames, such that a certain percentage of each frame is updated.

The correct choice of the INTRA percentage is a non-trivial issue that is influenced by the error characteristic of the channel, the decoder implementation, and the encoded video sequence. Obviously there is a trade-off to be considered. On the one hand, an increased percentage helps to reduce interframe error propagation. On the other hand, the coding efficiency is reduced at the same time. Besides the appropriate choice of INTRA percentage, there is great flexibility in the update pattern. Because the INTRA mode can be selected by the coding control of the encoder for each macroblock, it is up to the implementor to decide on the most effective scheme. For example, a certain number of INTRA coded macroblocks can be distributed randomly within a frame or clustered together in a small region.

In a very common scheme, which is also requested in H.263, each macroblock is assigned a counter that is incremented if the macroblock is encoded in interframe mode. If the counter reaches a threshold $T$, the macroblock is encoded in INTRA mode and the counter is reset to zero. By assigning a different initial offset to each macroblock, the update time of macroblocks can be distributed uniformly within the INTRA update interval $T$. However, the main reason for INTRA updates in H.263 is the *IDCT mismatch* rather than improved error resilience. IDCT mismatch may cause an accumulation of errors in the prediction loop if the encoder and decoder use slightly different implementations of the inverse DCT. Because this effect is usually small, a rather long update interval is sufficient. The H.263 standard requests a minimum value of $T = 132$. However, lower values are advantageous for error resilience. In our simulations we use a very similar update scheme and refer to it as *periodic INTRA update*. The only difference is that we also increment the counter for skipped macroblocks to guarantee a regular update of all image regions.

Other update schemes have been proposed in the literature to further improve the effectiveness of INTRA coding. In [59] and [68] it has been shown that it is advantageous to consider the image content when deciding on the frequency of INTRA coding. For example, image regions that cannot be concealed very well should be refreshed more often, whereas no INTRA coding is necessary for completely static background. This idea can also be included in a rate–distortion optimized encoding framework as proposed in [69] and [26], which is also discussed in more detail in Chapter 8, "Error Resilience Coding". Finally, different INTRA coding patterns, such as 9 randomly distributed macroblocks, 1×9, or 3×3 groups of macroblocks have been compared by Zhu and Kerofsky [25]. Though the shape of different patterns slightly influences the performance, the selection of the correct INTRA percentage has a significantly higher influence. We employ the *periodic INTRA update* scheme described above.

The results for the simulated transmission over a Rayleigh channel are presented in Fig. 12. From the DDF (left) it can seen that a choice of 6 % INTRA macroblocks results in the optimum performance for the Rayleigh fading channel at $E_b/N_0 = 18$ dB. Note that lower as well as higher percentages reduce the performance at a given value of PSNR$_e$. In particular, the loss in ΔPSNR is as much as 4 dB when comparing 6 % INTRA vs. 0 % INTRA at PSNR$_e$ = 36 dB. The right side of Fig. 12 shows that 6 % INTRA macroblocks is also the optimum choice for the whole range of channel conditions that are investigated. Therefore we will use this number in all following simulations. In fact, the results that we presented in the previous sections also use the described INTRA update scheme and 6 % INTRA macroblocks. Note, however, that the optimum INTRA percentage depends on the encoded sequence as well as the RWER and the concealment algorithm employed. In practice, it is therefore difficult to select the correct INTRA percentage.

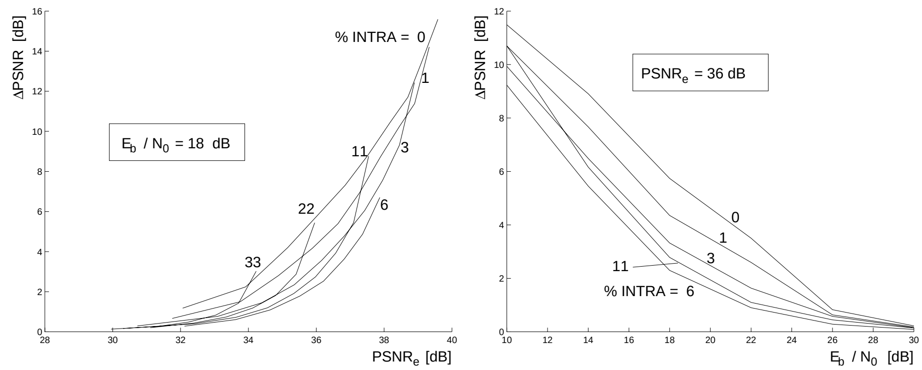


Figure 12: Evaluation of video quality using different amounts of INTRA coded macroblocks. Left: operational DDF for fixed channel. Right: ΔPSNR for fixed PSNR$_e$. The test sequence is *Mother and Daughter*.

### 4.4.3 Error Confinement

Leaky prediction and INTRA coding mainly address the problem of temporal error propagation. However, interframe error propagation also results in a spatial error spreading, as illustrated in Fig. 10. Due to motion-compensated prediction, errors may move from their original location to other parts of the frame and may even spread over the whole image. In some situations it is easier to combat the error by *confining* it to a well-defined sub-region of the frame. This can be achieved by restricting the range of motion vectors, such that no information outside the sub-region is used for prediction. In effect, the video sequence is partitioned into *sub-videos* that can be decoded independently.

Though the restriction of motion vectors could be guaranteed by the coding control of the encoder without any additional changes to the syntax, H.263 offers an optional mode for such a sub-video technique to allow a clearer and easier implementation. The *Independent Segment Decoding mode* (ISD mode) is described in Annex R of H.263. It can also be combined with the *Slice Structure mode*, but we restrict the discussion to the case in which a segment is identical to a GOB (assuming GOB-headers for each GOB). In the ISD mode, each GOB is encoded as an individual sub-video independently from other GOBs. All GOB boundaries are treated just like picture boundaries. This approach significantly reduces the efficiency of motion compensation, particularly for vertical motion, since image content outside the current GOB must not be used for prediction. To reduce the loss of coding efficiency the ISD mode is therefore often combined with the *Unrestricted Motion Vector mode* (UMV mode) which allows motion vectors pointing outside the coded picture area by extrapolating the image (or sub-video) borders. In spite of the UMV mode, typical losses in PSNR in the range from 0.2 to 2.0 dB often have to be accepted.

In case of transmission errors, the ISD mode assures that errors inside a GOB will not propagate to other GOBs, as illustrated in Fig. 13. Of course, the ISD mode alone does not solve the problem of temporal error propagation. It only simplifies keeping track of the error effects. However, the use of INTRA coding can also be used within each sub-video with very similar results. Because the ISD mode alone does not influence the overall performance significantly, we do not provide specific wireless video simulation results. Instead, we use this mode in combination with other modes described below.

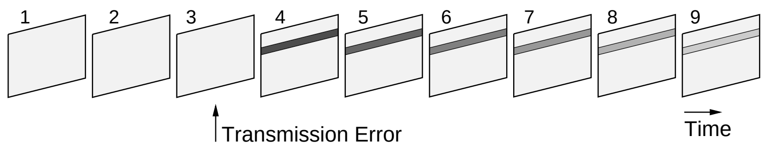


Figure 13: Illustration of spatio-temporal error propagation when the *Independent Segment Decoding mode* is used.

## 4.5 Feedback Based Error Control

As shown in the previous section, the remaining distortion after error concealment of corrupted image regions may be visible in the video sequence for several seconds. Although INTRA update schemes, as discussed above, help to limit the duration of error propagation, they also reduce coding efficiency. Furthermore, the optimum INTRA percentage is difficult to select in practice. In this section we discuss error resilience techniques that overcome these problems by utilizing a feedback channel from the receiver to the transmitter. As this approach assumes a special system architecture, we treat it in a separate section. The reader is also referred to Chapter 10 ("Feedback of Rate and Loss Information") for further techniques that utilize a feedback channel.

In our context, the feedback channel indicates which parts of the bit-stream were received intact and/or which parts of the video signal could not be decoded and had to be concealed. Depending on the desired error behavior, negative acknowledgment (NAK) or positive acknowledgment (ACK) messages can be sent. Typically, an ACK or NAK refers to a series of macroblocks or an entire GOB. NAKs require a lower bit-rate than ACKs, since they are only sent when errors actually occur, while ACKs have to be sent continuously. In either case, the required bit-rate is very modest compared to the video bit-rate of the forward channel. The feedback message is usually not part of the video syntax but transmitted in a different layer of the protocol stack where control information is exchanged. For example, in conjunction with H.263, ITU-T Recommendation H.245 [70] allows reporting of the temporal and spatial location of MBs that could not be decoded successfully and had to be concealed. Since the information is transmitted using a retransmission protocol, the error-free reception is guaranteed. However, additional delay may be introduced in the case of errors. In the following we assume that ACKs/NAKs are received reliably after a relatively large round trip delay of 300 ms.

### 4.5.1 Error Tracking

The Error Tracking approach uses the INTRA mode for selected MBs to stop interframe error propagation but limits its use to severely affected image regions only. During error-free transmission, the more effective INTER mode is used and the system therefore adapts effectively to varying channel conditions. This is accomplished by processing the NAKs from a feedback channel in the coding control of the encoder. Based on the information of a NAK, the encoder can reconstruct the interframe error propagation at the decoder. The coding control of a forward-adaptive encoder can then effectively stop interframe error propagation by avoiding the reference to severely affected MBs, e.g., by selecting the INTRA mode. If error concealment is successful and the error of a certain MB is small, the encoder may decide that INTRA coding is not necessary. For severe errors, a large number of MBs is encoded in INTRA mode, and the encoder may have to use a coarser quantizer to maintain a constant frame-rate and bit-rate. In this case, the overall picture quality at the source encoder decreases with a higher frequency of NAKs. Unlike retransmission techniques such as ARQ, Error Tracking does not increase the delay between encoder and decoder. It is therefore particularly suitable for applications that require a short latency.

Fig. 14 illustrates Error Tracking for the same example as in Fig. 10. As soon as the NAK is received with a system-dependent round-trip delay, the impaired MBs are determined and error propagation can be terminated by INTRA coding these MBs (frames 7-9). A longer round-trip delay just results in a later start of the error recovery. Note that it is necessary to track the shifting location of the errors to stop error propagation completely. In order to reconstruct the interframe error propagation that has occurred at the decoder, the encoder could store its own output bit-stream and decode it again, taking into account the reported loss of GOBs. While this approach is not feasible for a real-time implementation, it illustrates that the encoder, in principle, possesses all the information necessary to reconstruct the spatio-temporal error propagation at the decoder, once the NAKs have been received. For a practical system, the interframe error propagation has to be estimated with a low-complexity algorithm, as described in [71], [14], and [15] for a macroblock-based coder, such as H.263. A worst case estimate of error propagation that does not consider the severeness of errors is proposed by Wada [72].

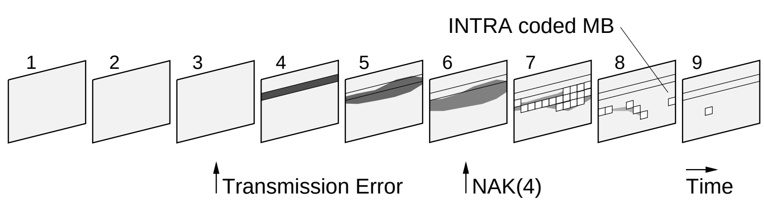


Figure 14: Illustration of spatio-temporal error propagation when *Error Tracking* is used.

The basic idea of the low-complexity algorithm is to carry out Error Tracking with macroblock resolution rather than pixel resolution. This is sufficient since the INTRA/INTER mode decision at the coder and the error concealment decision at the decoder are carried out for entire MBs as well. In a cyclical buffer for all MBs of the last several frames, the spatial overlap of MBs in successive frames due to motion-compensated prediction is stored, along with the error energy that would be introduced if concealment had to be used. If a NAK is received that indicates an error a few frames back, this error energy is "released" and "ripples" through the directed graph of frame-to-frame dependencies to the macroblocks of the current frame. Since all calculations are carried out on the MB level, the computational burden and memory requirements are small compared to the actual encoding of the video. For example, at QCIF resolution, there are only 99 MBs in each frame, as opposed to 38,016 luminance and chrominance samples.

Error Tracking is particularly attractive since it does not require any modifications of the bit-stream syntax of the motion-compensated hybrid coder. It is therefore fully compatible with standards such as H.261, H.263, or MPEG. The ITU-T recommends using previous frame concealment and Error Tracking with baseline H.263 and has included an informative appendix (Appendix 1) with Recommendation H.263. In addition, minor extensions of the H.245 control standard were adopted to include the appropriate NAK messages.

Fig. 15 shows the evaluation of the error tracking approach using the described simulation environment. For comparison, the *periodic INTRA update* scheme using 1 and 6 % INTRA macroblocks are included in the figures. Even when the optimum INTRA percentage is selected by the encoder (which is difficult in practice), the channel-adaptive Error Tracking shows significant gains. Note that Error Tracking would provide additional gains for round trip delays shorter than the 300 ms assumed here.

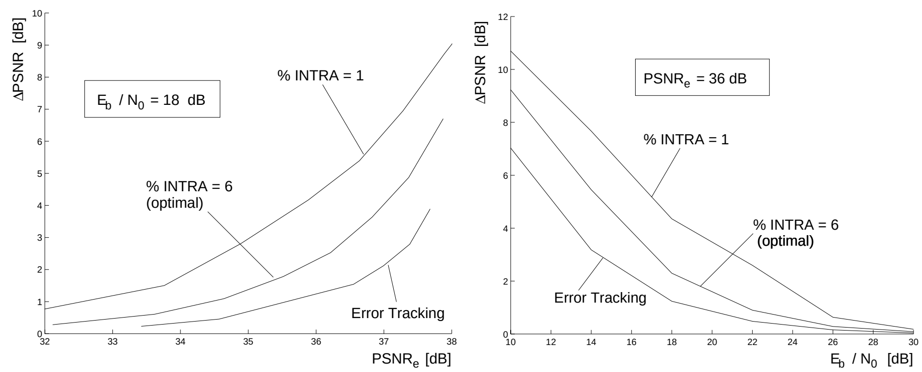


Figure 15: Evaluation of video quality using feedback information from the decoder and *Error Tracking*. Left: operational DDF for fixed channel. Right: ΔPSNR for fixed $\mathrm{PSNR}_e$. The test sequence is *Mother and Daughter*.

### 4.5.2 Reference Picture Selection

Rather than switching to INTRA mode at the encoder to stop interframe error propagation at the decoder, the encoder could also predict the current frame with reference to a previous frame that has been successfully decoded. This *Reference Picture Selection* (RPS) approach can lower the excess bit-rate due to NAK-induced INTRA coding [17] [73]. It is also described in Chapter 8, "Error Resilience Coding".

H.263+ has included RPS as an option, described in Annex N. As for the discussion of the ISD mode, we again consider the case that GOBs are used and each GOB starts with a header. Then, in H.263, the reference picture is selected on a GOB basis, i.e., for all MBs within one GOB the same reference picture is used. Future versions of H.263 are likely to contain an enhanced reference picture selection mode on the basis of MBs. In order to stop error propagation while maintaining the best coding efficiency, the last frame available without errors at the decoder should be selected. The RPS mode can be combined with the ISD mode for error confinement, or, for better coding efficiency, with an Error Tracking algorithm.

Reference Picture Selection can be operated in two different modes. When the encoder receives only *negative* acknowledgments, the operation of the encoder is not altered during error-free transmission, and the GOBs of the previous frame are used as a reference. After a transmission error, the decoder sends a NAK for an erroneous GOB and thereby requests that older, intact frames provide the reference-GOB. The typical transmission error effects are illustrated in Fig. 16, where the selection of reference-GOBs is indicated by arrows. Note that the use of the ISD mode is assumed and the indicated selection is only valid for the erroneous GOB. The encoder receives a NAK for frame 4 before the encoding of frame 7. The NAK includes the explicit request to use frame 3 for prediction, which is observed by the encoder. Similar to the Error Tracking approach, the quality degrades until the requested GOB arrives at the decoder, i.e., for the period of one round trip delay. Therefore, the loss of picture quality after a transmission error and the recovery after receiving a NAK behaves very similarly to basic Error Tracking. The advantage of the RPS mode vs. simply switching to INTRA mode lies in the increased coding efficiency. Fewer bits are needed for encoding the motion-compensated prediction error than for the video signal itself, even if the time-lag between the reference frame and the current frame is several frame intervals.

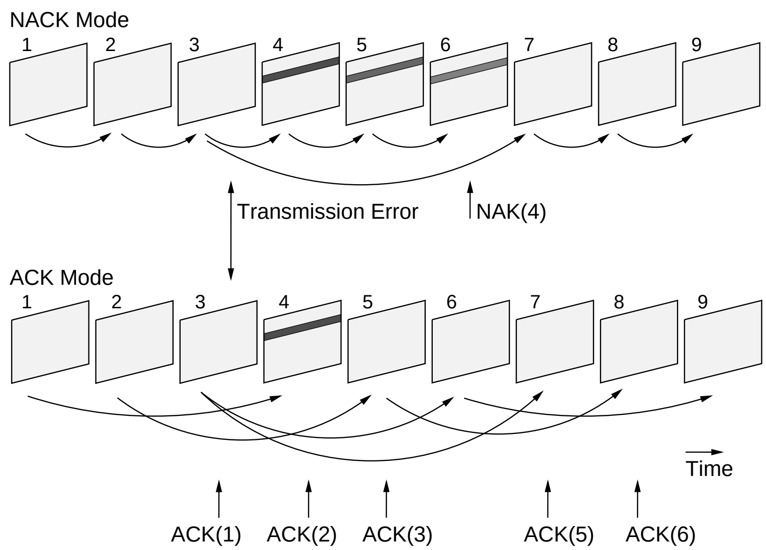


Figure 16: Illustration of spatio-temporal error propagation when the *Reference Picture Selection* is used.

In the *positive* acknowledgment mode, all correctly received GOBs are acknowledged and the encoder uses only those GOBs as a reference. Since the encoder has to use older reference pictures for motion-compensated prediction with increasing round-trip time, the coding performance decreases, even if no transmission errors occur. On the other hand, error propagation is avoided entirely since only error-free pictures are used for prediction.

Reference Picture Selection requires additional frame buffers at the encoder and decoder to store enough previous frames to cover the maximum round trip delay of NAKs or ACKs. In the NAK mode, the storage requirements of the decoder can be reduced to two frame buffers. Furthermore, if only error-free GOBs shall be displayed, one frame buffer is sufficient. In the ACK mode no such storage reduction is possible, unless a combination of both modes is allowed [73]. Increased storage requirements may still pose a problem for inexpensive mobile terminals for some time. Beyond that, there are proposals for increased coding efficiency in low bit-rate video codecs which use several or even many previous frames for prediction [74] [75] [76]. When using RPS, the additional frames can also be used to simultaneously increase error robustness [77].

For experimental evaluation of H.263 using reference picture selection, we use a combination of the RPS, ISD, and UMV modes. We assume that only NAKs are returned and use the same round trip delay as in the previous section, i.e., 300 ms. Even though the UMV mode helps to reduce the loss in coding efficiency that is caused in the ISD mode by restricted prediction, we observe that the overall loss in coding efficiency compared to the baseline mode is still considerable. Therefore, when comparing with Error Tracking, the overall gain by avoiding INTRA coding is diminished. For the test sequence shown we even observe that Error Tracking actually performs better as can be seen from Fig. 17. However, the difference is not significant and the order might be reversed for other test sequences. We therefore conclude that both feedback-based approaches perform equally well, and offer significant gains over error resilience schemes without feedback.

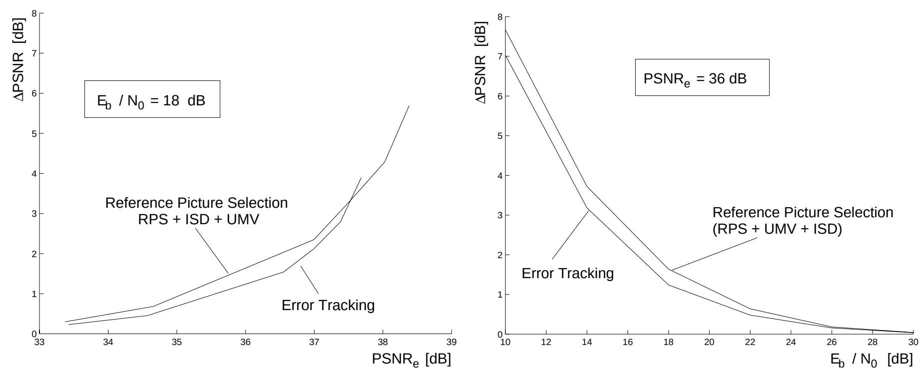


Figure 17: Evaluation of video quality using feedback information from the decoder and RPS. Left: operational DDF for fixed channel. Right: ΔPSNR for fixed $\mathrm{PSNR}_e$. The test sequence is *Mother and Daughter*.

# 5 Conclusions and Open Problems

In this chapter we have discussed the transmission of video over wireless channels with an emphasis on the interaction and trade-offs between system components. For evaluation, two types of distortions in the decoded video signal have to be considered, i.e., distortions due to source coding and distortions caused by transmission errors. Altering the bit allocation between source and channel coding usually has opposite effects on both distortion types, i.e., one increases while the other is reduced. To represent this trade-off formally we have introduced the *Distortion-Distortion Function* (DDF) which can be used as a tool for comparing wireless video systems.

For wireless video there are two major problems directly related to the transmission medium. (1) Only low bit-rates are available due to the limited availability of radio spectrum. (2) Path loss and multipath fading cause time-variant error rates. In designing the digital transmission system, there is a fundamental trade-off between *throughput*, *reliability*, and *delay*. The actual amount by which, e.g., reliability can be increased for a reduction in throughput, depends on the components of the *error control channel*, i.e., the channel, the modem, and the channel codec. This trade-off has been illustrated for the mobile radio channel assuming Rayleigh fading, BPSK modulation, and RS block codes. Especially for the mobile radio channel, the underlying physical mechanisms result in fundamental performance limits that require special measures in the video codec as the "last line of defense".

In Section 4, we have discussed such measures for low bit-rate video, with particular emphasis on techniques with relevance to the H.263 Recommendation. These error resilience techniques fall into two major categories – techniques that reduce the amount of introduced errors for a given error event (*resynchronization*, *error concealment*) and techniques that limit interframe error propagation (*leaky prediction*, *intra update*). The influence of both categories on the overall performance can be significant, as has been demonstrated using operational DDFs. For example, for a given PSNR at the encoder, the loss in PSNR caused by transmission errors can be reduced from more than 10 dB to less than 2 dB when combining existing options appropriately.

Additional gains can be obtained when using channel-adaptive source coding. The use of feedback information on the source coding level is an efficient and elegant technique for real-time multimedia communication over networks with heterogeneous QoS. Feedback schemes, such as *Error Tracking* or *Reference Picture Selection*, are suitable for interactive, individual communications, but they have inherent limitations outside this domain. They are particularly powerful if the round-trip delay is short. If the round-trip delay increases they become less efficient and ultimately useless. Also, feedback schemes rapidly deteriorate with increasing number of users. Their strength are point-to-point communications, and extensions to few users are possible at a loss in performance.

There are several other approaches to error resilient video transmission that could not be covered in this chapter. In particular, data partitioning and scalable video coding are of increased importance when feedback from the channel is not available. Instead, these approaches rely on priority mechanisms that are part of the network congestion management or provided implicitly through unequal error protection. Such techniques scale well with the number of users and are suitable for multicasting and broadcasting. On the other hand, they are less efficient than feedback schemes for point-to-point communications. Hybrid techniques that combine the advantages of channel-adaptive source coding based on feedback and scalable coding are not known today. Data partitioning and scalable video coding are covered in more detail in Chapter 7, "Layered Coding". These techniques can be very effective in practice.

Also, so-called "multiple-description" coding (MDC) schemes tailored towards graceful degradation for randomly selected subsets of the bitstream are in their infancy today. The multiple-description coding problem in its simplest form attempts to transmit a signal to three receivers, where receiver no. 1 receives a first bitstream, receiver no. 2 a second bitstream, and a third receiver both bitstreams. Such scenarios are directly relevant to broadcasting systems and transmission over erasure channels or best effort networks. Since Shannon's separation principle for source and channel coding does not hold, even the fundamental information theoretic performance limits for such schemes are unknown, let alone efficient algorithms that perform close to these limits. The situation becomes even more complicated, if the influence of a delay-constraint for real-time transmission is considered. More information on MDC is also included in Chapter 8, "Error Resilience Coding".

Besides the investigation of new coding schemes, the joint consideration of network and application layers will play an important role in the future. The close interaction between these layers may seem to require a vertical system integration in a closed network architecture. However, exactly the opposite architecture is becoming more and more important – horizontal integration with an IP-like spanning middle layer for seamless communication across network boundaries. Any efforts that require or assume closed network architectures are likely to be irrelevant in the future. Instead, the same interface/protocol will have to work across the wired and the wireless portion of the future multimedia network. There are two ways for multimedia applications to evolve within an open system architecture. On the one hand, end-to-end transport and error control for video/multimedia can be implemented in the terminals. On the other hand, the IP protocol stack can be refined and extended to offer additional services and functionalities that are required for efficient real-time transmission. Such multimedia-aware transport protocols need to consider the implications of wireless channels as the weakest link in the transmission chain. Otherwise, unnecessary performance degradation cannot be avoided. Judging from current efforts, both possibilities – end-to-end error control and multimedia-aware extensions of current transport protocols – will continue to evolve and contribute to the ultimate success of wireless video.

# References


[1] J. Uddenfeldt, "Digital Cellular - Its Roots and Its Future," Special Issue on Mobile Radio Centennial, *Proceedings of the IEEE*, vol. 86, no. 7, pp. 1319-1324, July 1998.

[2] B. Girod, N. Färber, and E. Steinbach, "Performance of the H.263 Video Compression Standard," *Journal of VLSI Signal Processing: Systems for Signal, Image, and Video Technology*, vol. 17, pp. 101-111, Nov. 1997.

[3] E. Berruto, M. Gudmonson, R. Menolascino, W. Mohr, and M. Pizarroso, "Research Activities on UMTS Radio Interface, Network Architectures, and Planning," *IEEE Commun. Mag.*, vol. 36, no. 2, pp. 82-95, Feb. 1998.

[4] D. Grillo (ed.), Special Section on Third-Generation Mobile Systems in Europe, *IEEE Personal Commun. Mag.*, vol. 5, no. 2, pp. 5-38, April 1998.

[5] P. Bahl and B. Girod (eds.), Special Section on Wireless Video, *IEEE Commun. Mag.*, vol. 36, no. 6, pp. 92-151, June 1998.

[6] S. Lin, D.J. Costello, and M.J. Miller, "Automatic repeat error control schemes," *IEEE Commun. Mag.*, vol. 22, pp. 5-17, Dec. 1984.

[7] A. Heron and N. MacDonald, "Video transmission over a radio link using H.261 and DECT," in *Inst. Elect. Eng. Conf. Publications*, no. 354, pp. 621-624, 1992.

[8] M. Khansari, A. Jalali, E. Dubois, and P. Mermelstein, "Low Bit-Rate Video Transmission over Fading Channels for Wireless Microcellular Systems," *IEEE Trans. on Circuits and Systems for Video Technol.*, vol. 6, no. 1, pp. 1-11, Feb. 1996.

[9] Y. Wang and Q.-F. Zhu, "Error Control and Concealment for Video Communication: A Review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974-997, May 1998.

[10] R. Fischer, P. Mangold, R.M. Pelz, and G. Nitsche, "Combined Source and Channel Coding for Very Low Bitrate Mobile Visual Communication Systems," in Proc. *Int. Picture Coding Symposium (PCS)*, Melbourne, Australia, pp. 231-236, March 1996.

[11] K. Illgner and D. Lappe, "Mobile multimedia communications in a universal telecommunications network," in Proc. *Visual Communications and Image Processing (VCIP)*, Taipei, Taiwan, SPIE vol. 2501, pp. 1034-1043, May 1995.

[12] 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 (eds.), *Multimedia Communications and Video Coding*, pp. 475-482, New York: Plenum Press, 1996.

[13] R. Talluri, "Error-Resilient Video Coding in the MPEG-4 Standard," *IEEE Commun. Mag.*, vol. 36, no. 6, pp. 112-119, June 1998.

[14] E. Steinbach, N. Färber, and B. Girod, "Standard Compatible Extension of H.263 for Robust Video Transmission in Mobile Environments," *IEEE Trans. on Circuits and Systems for Video Technol.*, vol. 7, no. 6, pp. 872-881, Dec. 1997.

[15] B. Girod, N. Färber, and E. Steinbach, "Error-Resilient Coding for H.263," in D. Bull, N. Canagarajah, A. Nix (eds), *Insights into Mobile Multimedia Communication*, pp. 445-459, Academic Press, New York, 1999.

[16] S. Wenger, G. Knorr, L. Ott, and F. Kossentini, "Error Resilience Support in H.263+," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 8, no. 7, pp. 867-877, Nov. 1998.

[17] S. Fukunaga, T. Nakai, and H. Inoue, "Error resilient video coding by dynamic replacing of reference pictures," in Proc. *IEEE Global Telecomun. Conf. (GLOBECOM)*, vol. 3, London, GB, pp. 1503-1508, Nov. 1996.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiley & Sons, New York, 1991.

[19] G. Buch, F. Burkert, J. Hagenauer, and B. Kukla, "To compress or not to compress?," in Proc. *Commun. Theory Conf. in conj. w. IEEE Global Telecomun. Conf. (GLOBECOM)*, pp. 198-203, London, UK, Nov. 1996.

[20] H. Jafarkhani and N. Farvardin, "Channel-matched hierarchical table-lookup vector quantization for transmission of video over wireless channels," in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, Lausanne, Switzerland, vol. 3, pp. 755-758, Sept. 1996.

[21] R.E. Van Dyck and D.J. Miller, "Transport of Wireless Video Using Separate, Concatenated, and Joint Source-Channel Coding," *Proceedings of the IEEE*, vol. 87, no. 10, pp. 1734-1750, Oct. 1999.

[22] B. Belzer, J. Villasenor, B. Girod, "Joint source and channel coding of images with trellis coded quantization and convolutional codes," in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, Washington, D.C., vol. 2, pp. 85-88, Oct. 1995.

[23] S. B. Wicker, *Error Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[24] B. Girod, "Psychovisual Aspects of Image Communication," *Signal Processing*, vol. 28, no. 3, pp. 239-251, Sept. 1992.

[25] Q. F. Zhu and L. Kerofsky, "Joint source coding, transport processing, and error concealment for H.323-based packet video," in Proc. *Visual Commun. and Image Processing (VCIP)*, San Jose, CA, USA, SPIE vol. 3653, pp. 52-62, January 1999.

[26] S. Wenger, G. Côté, "Using RFC2429 and H.263+ at low to medium bit-rates for low-latency applications," in Proc. *Packet Video Workshop (PVW)*, New York, NY, USA, April 1999.

[27] A. Ortega and K. Ramchandran, "Rate-Distortion Methods for Image and Video Compression," *IEEE Signal Processing Mag.*, vol. 15, no. 6, pp. 23-50, Nov. 1998.

[28] B. Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems, Part I: Characterization," *IEEE Commun. Mag.*, vol. 35, no. 9, pp. 136-146, Sept. 1997.

[29] W. C. Jakes, *Microwave Mobile Radio Reception*, New York: Wiley, 1974.

[30] P. Wong and D. Britland, *Mobile Data Communications Systems*, Artech House, Norwood MA, 1995.

[31] J. G. Proakis, *Digital Communications*, 2d ed., McGraw-Hill, New York, 1989.

[32] B. Sklar, *Digital Communications, Fundamentals and Applications*, Englewood Cliffs, Prentice Hall, NJ, 1988.

[33] J. E. Padgett, C. Günther, and T. Hattori, "Overview of Wireless Personal Communications," *IEEE Commun. Mag.,* vol. 33, no. 1, pp. 28-41, Jan. 1995.

[34] R. Cam and C. Leung, "Throughput Analysis of Some ARQ Protocols in the Presence of Feedback Errors," *IEEE Trans. on Commun.*, vol. 45, no. 1, pp. 35-44, Jan. 1997.

[35] M. Zorzi, R. Rao, and L.B. Milstein, "Error Statistics in Data Transmission over Fading Channels," *IEEE Trans. on Commun.*, vol. 46, no. 11, pp. 1468-1477, Nov. 1998.

[36] Y. Yao, "An Effective Go-Back-N ARQ Scheme for Variable-Error-Rate Channels," *IEEE Trans. on Commun.*, vol. 43, no. 1, pp. 20-23, Jan. 1995.

[37] C. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video tranmission over wireless channels," in Proc. *Visual Commun. and Image Processing (VCIP)*, San Jose, CA, USA, SPIE vol. 3024, pp. 1200-1211, Feb. 1997.

[38] C. Hsu, A. Ortega, and M. Khansari, "Rate control for robust video tranmission over Burst-Error Wireless Channels," *IEEE J. on Selected Areas in Commun.*, vol. 17, no. 5, pp. 756-772, May 1999.

[39] M. Podolsky, S. McCanne, and M. Vetterli, "Soft ARQ for Layered Streaming Media," Tech. Report, no. UCB/CSD-98-1024, Computer Science Division, University of California, Berkley, CA, Nov. 1998.

[40] N. Färber and B. Girod, "Robust H.263 Compatible Video Transmission for Mobile Access to Video Servers," in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, Santa Barbara, CA, USA, vol. 2, pp. 73-76, Oct. 1997.

[41] P. Cherriman and L. Hanzo, "Programable H.263-Based Video Transceivers for Interference-Limited Environments," *IEEE Trans. on Circuits and Systems for Video Technol.*, vol. 8, no. 3, pp. 275-286, June 1998.

[42] T. Gardos, "H.263+: The New ITU-T Recommendation for Video Coding at Low Bit Rates," in Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Seattle, WA, USA, vol. 6, pp. 3793-3796, May 1998.

[43] G. Côté, B. Erol, M. Gallant, and F. Kossentini, "H.263+: Video Coding at Low Bit Rates," *IEEE Trans. on Circuits and Systems for Video Technol.*, vol. 8, no. 7, pp. 849-866, Nov. 1998.

[44] Telenor Research, Video codec test model, TMN5, http://www.nta.no/brukere/DVC, 1995, PD software.

[45] J. Huber and A. Rüppel, “Zuverlässigkeitsschätzung für die Ausgangssymbole von Trellis-Decodern,” *Archiv für Elektronik und Übertragungstechnik (AEÜ)*, vol. 44, no. 1, pp. 8-21, Jan. 1990.

[46] J. Hagenauer, and P. Höher, “A Viterbi algorithm with soft-decision output and its applications”, in Proc. *IEEE Global Telecommun. Conf., (GLOBECOM)*, Dallas, TX, USA, pp. 47.1.1-47.1.7, Nov. 1989.

[47] C. Chen, “Error Detection and Concealment with an Unsupervised MPEG2 Video Decoder,” *J. of Visual Commun. and Image Representation*, vol. 6, no. 3, pp. 265-278, Sep. 1995.

[48] W.-M. Lam and A. R. Reibman, “An Error Concealment Algorithm for Images Subject to Channel Errors,” *IEEE Trans. Image Processing*, vol. 4, no. 5, pp. 533-542, May. 1995.

[49] J. W. Park, J. W. Kim, and S. U. Lee, “DCT Coefficients Recovery-Based Error Concealment Technique and Its Application to the MPEG-2 Bit Stream,” *IEEE Trans. Circuits and Sys. for Video Technol.*, vol. 7, no. 6, pp. 845-854, Dec. 1997.

[50] T. J. Ferguson and J. H. Rabinowitz, “Self-synchronizing Huffman codes,” *IEEE Trans. Inform. Theory*, vol. IT-30, no. 4, pp. 687-693, July 1984.

[51] W.-M. Lam and A. R. Reibman, “Self-synchronizing variable length codes for image transmission,” in Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, San Francisco, CA, USA, vol. 3, pp. 477-480, March 1992.

[52] J. Wen and J. D. Villasenor, “A Class of Reversible Variable Length Codes for Robust Image and Video Coding,” in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, Santa Barbara, CA, USA, vol. 2, pp. 65-68, Oct. 1997.

[53] J. Wen and J. D. Villasenor, “Reversible Variable Length Codes for Efficient and Robust Image and Video Coding,” in Proc. *Data Compression Conference (DCC)*, Snowbird, Utah, USA, pp. 471-480, March 1998.

[54] B. Girod, “Bidirectionally Decodable Streams of Prefix Code Words,” *IEEE Commun. Letters*, vol. 3, no. 8, pp. 245-247, Aug. 1999.

[55] D. W. Redmill and N. G. Kingsbury, “The EREC: An Error-Resilient Technique for Coding Variable-Length Blocks of Data,” *IEEE Trans. Image Processing*, vol. 5, pp. 565-574, Apr. 1996.

[56] Y. Wang, Q.-F. Zhu, and L. Shaw, “Maximally smooth image recovery in transform coding,” *IEEE Trans. Commun.*, vol. 41, pp. 1544-1551, Oct. 1993.

[57] H. Sun and W. Kwok, “Concealment of damaged block transform coded images using projection onto convex sets,” *IEEE Trans. Image Processing*, vol. 4, pp. 470-477, Apr. 1995.

[58] H. Sun and J. Zedepski, “Adaptive Error Concealment Algorithm for MPEG Compressed Video,” in Proc. *Visual Commun. and Image Processing (VCIP)*, Boston, MA, USA, SPIE vol. 1818, pp. 814-824, Nov. 1992.

[59] P. Haskell and D. Messerschmitt, “Resynchronization of motion compensated video affected by ATM cell loss,” in Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, San Francisco, CA, USA, vol. 3, pp. 545-548, March 1992.

[60] W.-M. Lam, A.R. Reibman, and B. Lin, "Recovery of lost or erroneously received motion vectors," in Proc. *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Minneapolis, MN, USA, vol. 5, pp. 417-420, April 1993.

[61] P. Salama, N. Shroff, and E. J. Delp, "A Fast Suboptimal Approach to Error Concealment in Encoded Video Streams," in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, Santa Barbara, CA, USA, vol. 2, pp. 101-104, Oct. 1997.

[62] N.S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice Hall, Englewood Cliffs, New Jersey, 1984.

[63] D. J. Connor, "Techniques for Reducing the Visibility of Transmission Errors in Digitally Encoded Video Signals," *IEEE Trans. on Commun.*, vol. COM-21, no. 6, pp. 695-706, June 1973.

[64] B. Girod, "The Efficiency of Motion-Compensating Prediction for Hybrid Coding of Video Sequences," *IEEE Journal on Selected Areas in Commun.*, vol. SAC-5, no. 7, pp. 1140 - 1154, August 1987.

[65] B. Girod, "Motion-Compensating Prediction with Fractional-Pel Accuracy," *IEEE Trans. on Commun.*, vol. 41, no. 4, pp. 604 - 612, April 1993.

[66] B. Girod and N. Färber, "Feedback-Based Error Control for Mobile Video Transmission," Special Issue on Video for Mobile Multimedia, *Proceedings of the IEEE*, vol. 97, no. 10, pp. 1707-1723, Oct. 1999.

[67] N. Färber, K. Stuhlmüller, and B. Girod, "Analysis of Error Propagation in Hybrid Video Coding with Application to Error Resilience," in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, Kobe, Japan, vol. 2, pp. 550-554, Oct. 1999.

[68] J. Liao and J. Villasenor, "Adaptive Intra Update for Video Coding over Noisy Channels," in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, Lausanne, Switzerland, vol. 3, pp. 763-766, Sept. 1996.

[69] T. Wiegand, Personal Communication, July 1998.

[70] ITU-T Recommendation H.245, "Control Protocol for Multimedia Communication," 1996.

[71] N. Färber, E. Steinbach, and B. Girod, "Robust H.263 Compatible Video Transmission Over Wireless Channels," in Proc. *Int. Picture Coding Symposium (PCS)*, Melbourne, Australia, pp. 575-578, March 1996.

[72] M. Wada, "Selective recovery of video packet loss using error concealment," *IEEE J. on Select. Areas in Commun.*, vol. 7, no. 5, pp. 807-814, June 1989.

[73] Y. Tomita, T. Kimura, and T. Ichikawa, "Error Resilient Modified Inter-Frame Coding System for Limited Reference Picture Memories," in Proc. *Int. Picture Coding Symposium (PCS)*, Berlin, Germany, pp. 743-748, Sept. 1997.

[74] T. Wiegand, X. Zhang, and B. Girod, "Motion-Compensating Long-Term Memory Prediction," in Proc. *IEEE Int. Conf. on Image Processing (ICIP)*, Santa Barbara, CA, USA, vol. 2, pp. 53-56, Oct. 1997.

[75] T. Wiegand, X. Zhang, and B. Girod, "Long-Term Memory Motion-Compensated Prediction," *IEEE Trans. on Circuits and Systems for Video Technol.*, vol. 9, no. 1, pp. 70-84, Feb. 1999.

[76] B. Girod, T. Wiegand, E. Steinbach, M. Flierl, and X. Zhang, "High-Order Motion Compensation for Low Bit-Rate Video," in Proc. *European Signal Processing Conference (EUSIPCO)*, Island of Rhodes, Greece, vol. 1, pp. 253-256, Sept. 1998, Invited Paper.

[77] T.Wiegand, N. Färber, and B. Girod, "Error-Resilient Video Transmission Using Long-Term Memory Motion-Compensated Prediction," *IEEE J. on Selected Areas in Commun.*, accepted.