ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FUCULITY OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Source-Channel Prediction for Error Resilient Video Coding

A thesis submitted to the school of graduate studies in
Partial fulfillment of the requirements of
Masters of Science in Electrical and Computer Engineering
(Communication)

By

Kibrom Negash

Advisor: - Dr. Ing. Hailu Ayele

September 2005

Addis Ababa

ADDIS ABABA UNIVERSTY
SCHOOL OF GRADUATE STUDIES
FACILITY OF TECHNOLOGY

Source- Channel Prediction for Error Resilient Video coding

By
Kibrom Negash

Approval by board of examiners

______________________	______________________
Chairman, Department Graduate Committee	**Signature**
______________________	______________________
Adviser	**Signature**
______________________	______________________
Internal Examiner	**Signature**
______________________	______________________
External Examiner	**Signature**

ACKNOWLEDGMENT

I am deeply indebted to Dr.Ing. Hailu Ayele, my advisor, with out whom I could never have accomplished the result of the thesis work. I would like also to thank all my friends who helped me while running the video codec software. Finally, I am very much indebted to the postgraduate computer laboratory staffs of Electrical and Computer Engineering at Addis Ababa University.

TABLE OF CONTENT

Acronym

ACK	- Acknowledgment
EC	- Motion compensation
FEC	- Forward error correction
DCT	- Discrete cosine transform
DMC	-Discrete memory less channel
GOP	- Group of block
IDCF	-inverse discrete cosine transfer
JPEG	- Joint picture expert group
MB	- Macro block
MPEG	- Moving picture expert group
MSE	- Mean square error
MV	- Motion vector
NAK	- Negative Acknowledgment
RLE	- Run length encoder
R-D function	-Rate–Distortion function
RTP	- Real time protocol
I-FRAME	- Intra frame
P-FRAME	- Inter frame
PSNR	- Peak signal to noise ratio
VLC	-variable length code

LIST OF FIGURES

ABSTRACT

Over recent decades the role of image in the communication of information has grown steadly. Advances in technologies underlying the capture, transfer, storage, and display of images as a means of communicating information has become technologically and economically feasible. More importantly, images are in many situations an extremely efficient means to communicate information.

The approach in this thesis work is to modify conventional motion compensation prediction so as to improve or optimize the error resilience performance of the over all system; motion compensated prediction in conventional video coders is based on the encoder reconstruction of the previous frame where motion is estimated so as to minimize the encoder prediction error. Such design paradigm optimizes the prediction at the encoder and ignores the effect of packet loss.

The thesis work is based on the literature surveys, the simulation results are based on the H.263 software and the necessary computation is done on MATLAB. The modified (source-channel prediction) scheme achieves better error resilience performance and better over all R-D trade off than its conventional counter part.

CHAPTER I

1.1 DISCRIPION OF THE PROJECT

Video information (perhaps audio as well) is transmitted over telecommunication links including networks, telephone lines, ISDN and radio. Video has high bandwidth (i.e. many bytes of information per second) and so these applications require video compression with video coding technology to reduce the bandwidth before transmission. In error resilient video coding, motion predication is based on the encoder reconstruction of the pervious frame where motion is estimated so as to minimize the encoder prediction error. Such design paradigms optimize the prediction at the encoder and ignores the effect of packet loss in the channel.

To transmit video over noisy channel, one uses both source and channel coding according to Shannon's separation principle, these components can be designed independently without loss in performance. However this important information theoretic result is based on several assumptions that might breakdown in practice, in particular it is based on: -

i) The assumption of an infinite block length for both source and channel coding; and

ii) An exact and complete knowledge of the statistics of transmission channel

As a result of the first assumption, the separation principle can't be applied with out performance loss to applications with real time constraints also, as a consequence of second assumption; it applies only to point-to-point communication. Therefore, joint source-channel coding and error resilient coding can be advantageous in practice and hence this work will be based on source-channel prediction in error resilient video coding.

The damage due to the packet loss is greatly exacerbated by error propagation. Many error resilient video coding techniques are focused on the prediction mechanism, for example H.263 standard divides a picture into non overlapping spatial regions, i.e. slices and limits spatial and temporal prediction with each slice. The work will modify conventional motion compensated prediction so as to improve or optimize the error resilience performance of the over all system through the source-channel prediction

The simulation system is based on H.263 codec standard. A sequence is encoded into H.263 bit stream; the bit stream undergoes a packet loss pattern that is randomly generated. The system performance is measured by the average luminance PSNR

1.2 POSSIBLE APPLICATION

Video conferencing and video telephony have a wide range of application including: -

- Desk top and room based conferencing
- Video over the internet and over telephone line
- Surveillance and monitoring
- Telemedicine (medical consultation and diagnosis at a distance)
- Computer based training and education

In each case visual information is transmitted over telecommunication link and the quality of the video at the receiving end needs to be as high as possible. The result of this work will help to avoid the effect of possible packet loss and have good picture quality at the receiving end.

CHAPTER II

DIGITAL COMMUNICATION SYSTEM

2.1 INTRODUCTION

A source for information generates a message example of which includes human voice, television picture, pressure, etc. In these examples, the message is not electrical in nature and so a transducer is used to convert it into electrical waveform called message signal. The waveform is also refereed to us as a base-band signal; the term base-band is used to designate the band of frequencies representing the massage signal generated at the source [6]. In a digital communication system, the messages produced by sources need to be converted to a sequence of binary digits. In the following chapter digital representation of analog signal will be discussed

2.2 SOURCE IN DIGITAL COMMUNICATION SYSTEM

An analog signal can always be converted to digital form by combing three basic operations: sampling, quantizing and encoding as shown in the block diagram of figure2.1

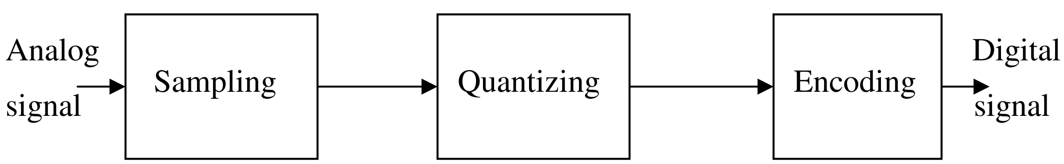


Figure 2.1 analogs to digital conversion

In sampling operation, only sample values of analog signal at uniformly spaced discrete instants of time are retained. In order to ensure good construction of the message at the receiver, the sampling rate must be greater than twice the highest frequency component W of the massage wave [5] [6]. In practice, a low-pass filter is used at the front end of the sampler in order to exclude frequencies greater than W before sampling. Thus the application of sampling permits the reduction of the continuously varying massage wave to a limited number of discrete values per second.

The conversion of an analog (continuous) sample of the signal into a discrete form is called quantizing processes. In quantizing operation, each sample value is approximated by the nearest level in a finite set of discrete level. In the encoding operation the selected level is represented by a codeword that consist of a prescribed number of code elements. As a result of sampling and quantizing operations, errors are introduced into the digital signal. These errors are nonreversible in that it is not possible to reproduce an exact replica of the original analog signal from its digital representation. Indeed, by proper selection of the sampling rate and code word length (i.e. number of quantization levels), the errors due to sampling and quantizing can be made so small that the difference between the analog signal and reconstructed digital signal is not discernible by a human observers [1] [5].

2.2.1 OPTIMUM QUANTIZATION

As discussed in the preceding paragraphs, an analog source emits a message wave form x(t) that is a sample function of a stochastic process X(t) is a band limited, stationary stochastic process. The sampling theorem allows us to represent x(t) by a sequence of uniform samples taken at the Nyquist rate[5]. By applying the sampling theorem, the output of the analog source is converted to an equivalent discrete time sequence sample. The samples are then quantized in amplitude and encoded. One type of sample encoding is to represent each discrete amplitude level by a sequence of binary digits. Hence, if we have L levels, we need $R = \log_2 L$ bits per sample if L is a power of 2 or $R = \lfloor \log_2 L \rfloor + 1$ if L is not a power of 2, where $\lfloor \log_2 L \rfloor$ is largest integer less than $\log_2 L$. On the other hand, if the levels are not equally probable, and probabilities of the output levels are known, we may use Huffman coding (also called entropy coding) to improve the efficiency of the encoding process [5] [10]. Quantization of the amplitude of the sampled signals results in data compression but it could also introduces some distortion of the waveform or a loss of signal fidelity. The minimization of this distortion is considered in this section. Many of the result given in the coming section apply directly to a discrete–time, continuous amplitude, memoryless Gaussian source such a source serves as a good model for the residual error in a number of source coding methods.

2.2.2 RATE DISTORTION FUNCTION

For the ease of mathematical derivation let us begin the discussion of the quantization by considering the distortion introduced when the samples from the information source are quantized to a fixed number or bits. By the term "distortion" is meant some measure of the difference between the actual source samples $\{x_k\}$ and the corresponding quantized value $\{\tilde{x}_k\}$[5], which is denoted by $d\{x_k, \tilde{x}_k\}$. For example, a commonly used distortion measure is the squared error distortion defined as

$$d(x_k, \tilde{x}_k) = (x_k - \tilde{x}_k)^2 \tag{2.1}$$

Other distortion measures may take the general form

$$d(x_k, \tilde{x}_k) = |x_k - \tilde{x}_k|^p \tag{2.2}$$

Where p takes values from the set of positive integers. The case p=2 has the advantage of being mathematically tractable. If $d(x_k, \tilde{x}_k)$ is the distortion measure per letter the distortion between sequences of n samples X_n and the corresponding n quantized value $\tilde{X}_n$ is the average over the n source output sample, i.e.

$$d(X_k, \tilde{X}_k) = \frac{1}{n}\sum_{k=1}^{n} d(x_k, \tilde{x}_k) \tag{2.3}$$

As the source output is random process, the n samples in X_n are random variables. The expected value of $d(X_n, \tilde{X}_n)$ is defined as the distortion D

$$D = E[d(X_n, \tilde{X}_n)] = \frac{1}{n}\sum_{k=1}^{n} E[d(x_k, \tilde{x}_k)] = E[d(x, \tilde{x})] \tag{2.4}$$

Where the last step follows from the assumption that the source output process is stationary. Considering a memoryless source with a continuous amplitude output X that has a pdf p(x), a quantized amplitude out alphabet $\tilde{X}$ and a per letter distortion measure $d(x_k, \tilde{x}_k)$, where x∈X and $\tilde{x} \in \tilde{X}$, then the minimum rate in bits per source output that is required to represent the output X of the memoryless source with distortion less than or equal to D is called the rate-distortion function R(D) and is defined as

$$R(D) = \min_{p(\tilde{x}/x):E[d(x,\tilde{x})]\leq D} I\left(X,\tilde{X}\right) \tag{2.5}$$

Where $I(X;\tilde{X})$ is the average mutual information between X and $\tilde{X}$. In general, the rate R(D) has inverse relationship with D. As the source is modeled as memory less Gaussian source, we have the following theorems on rate distortion function from Shannon,

"The minimum information rate necessary to represent the output of a discrete-time, continuous-amplitude memory less Gaussian source based on a mean–squared error distortion measure per symbol (single letter distortion measure) is

$$R_g(D) = \begin{cases} \frac{1}{2}\log_2\left(\sigma_x^2/D\right) & ;\ 0\leq D \leq \sigma_x^2 \\ 0 & ;\ D > \sigma_x^2 \end{cases} \tag{2.6}$$

Where σ_x^2-variance of the Gaussian source output."[5]. The above equation implies that no information need to be transmitted when a distortion $D \geq \sigma_x^2$. If the function dependence between D and R in the equation (2.6) is reversed, we express D in terms of R as

$$D_g(R) = 2^{-2R}\sigma_x^2 \tag{2.7}$$

$$10\log_{10} D_g(R) = -6R + 10\log_{10}\sigma_x^2$$

And the above function is known as the distortion rate function for discrete memory less source. Before concluding this section, let us consider a band limited Gaussian source with spectral density

$$\Phi(f) = \begin{cases} \frac{\sigma_x^2}{2w} & \text{if } |f| \leq W \\ 0 & \text{if } |f| > W \end{cases} \tag{2.8}$$

When the output of this source is sampled at the Nyquist rate, the samples are uncorrelated and, since the source is Gaussian, they are also statistically independent.

Hence, the equivalent discrete-time Gaussian source is memoryless. The rate distortion function for each sample is given by equation (2.6). Hence, the rate–distortion function for the band limited white Gaussian source in bits/sec is

$$R_g(D) = W \log_2\left(\sigma_x^{2}/D\right) \qquad (0 \le D \le \sigma_x^2)$$

$$D_g(R) = 2^{-2R/W}\sigma_x^{2} \qquad (2.9)$$

Which, when expressed in decibel and normalized by σ_x^2, becomes

$$10\log\left(Dg(R)/\sigma_x^{2}\right) = -2R/W \qquad (2.10)$$

Scalar quantization

In the source encoding, the quantizer can be optimized if the probability density function of the signal amplitude at the input to quantizer is known. Suppose that the sequence $\{x_n\}$ at the input to the quantizer has a PDF P(x) and let L=2^R be the desired number of levels. We wish to design the optimum scalar quantizer that minimizes some function of the quantization error $q = \tilde{x} - x$, where $\tilde{x}$ is the quantized value of x. To elaborate, suppose that $f(\tilde{x} - x)$ denotes the desired function of the error. Then, the average distortion resulting from quantization of the signal amplitude is:

$$D = \int_{-\infty}^{\infty} f(\tilde{x} - x)p(x)dx \qquad (2.11)$$

In general, an optimum quantizer is one that minimizes D by optimally selecting the output levels and the corresponding input range of each output level [5]. For a uniform quantizer, the output levels are specified as $\tilde{x}_k$ =1/2 (2k-1) Δ, corresponding to an input signal amplitude in the range (k-1)<x<kΔ, whereΔ is the step size. When the uniform quantizer is symmetric with an even number of levels, the average distortion in (2.11) may be expressed as:

$$D = 2\sum_{k=0}^{L/2-1}\int_{(k-1)\Delta}^{K\Delta} f\left(1/2(2k-1)\Delta - x\right)p(x)dx + 2\int_{-(L/2-1)\Delta}^{\infty} f'\left(1/2(2k-1)\Delta - x\right)p(x)dx \qquad (2.12)$$

In this case, the minimization of D is carried out with respect to the step-size parameter Δ. By differentiating D with respect to Δ, we obtain:

$$2\sum_{k=0}^{L/2-1} 2(2k-1)\int_{(k-1)\Delta}^{K\Delta} f\left(\tfrac{1}{2}(2k-1)\Delta - x\right)p(x)dx + (L-1)\int_{-(L/2-1)\Delta}^{\infty} f'\left(\tfrac{1}{2}(2k-1)\Delta - x\right)p(x)dx = 0$$

Where f'(x) denotes the derivative of f (x). By relaxing the constraint that the quantizer be uniform, the distortion can be reduced further. In this case, we let the output level be $\tilde{x} = \tilde{x}_k$ when the input signal amplitude is in the range $x_{k-1}<x<x_k$. For an L-level quantizer, the end points are x_0 =-∞ and x_L=∞, the resulting distortion is:

$$D = \sum_{K=1}^{L} \int_{x_k-1}^{x_k} f(\tilde{x} - x)p(x)dx \qquad (2.13)$$

which is now minimized by optimally selecting the { $\tilde{x}_k$ } and { x_k }

To conclude, the quantizer can be optimized when the pdf of the continuous source output is known. The optimum quantizer of $L=2^R$ levels result in a minimum distortion of D(R), where $R=\log_2 L$ bits/sample. Thus, this distortion can be achieved by simply representing each quantized sample by R bits [5] [6]. However more efficient encoding is possible. The discrete source output that results from quantization is characterized by a set of probabilities $\{P_k\}$ that can be used to design efficient variable length codes for the source output (entropy coding). The efficiency of any encoding method can be compared with the distortion rate function.

Vector Quantization

In the previous section, the quantization of the output signal from a continuous amplitude source when the quantization is performed on a sample-by-sample basis, i.e. by scalar quantization is considered. In this section, we consider the joint quantization of a block of signal samples or a block of signal parameters. This type of quantization is called block or vector quantization [6] [10].

A fundamental result of rate-distortion theory is that quantizing vectors instead of scalars can achieve better performance. If, in addition, the signal samples or signal parameters are statistically independent, we can exploit the dependency by jointly quantizing blocks of samples or parameters and, thus, achieve an even greater efficiency compared with that which is achieved by scalar quantization. The vector quantization problem may be formulated as follows. We have an n-dimensional vector $X = [X_1, X_2, ..., X_n]$ with real valued, continuous amplitude components { x_k , $1 \le k \le n$} that are described by a joint pdf $p(x_1, x_2, \ldots, x_n)$. The vector X is quantized into another n– dimensional vector $\tilde{X}$ with components. {$\tilde{x}_k$, $1 \le k \le n$} We express the quantization as Q(-) so that

$$\tilde{X} = Q(x)$$

Where $\tilde{X}$ is the output of vector quantizer when the input vector is X. In general, quantization of the n-dimensional vector X into an n-dimensional vector $\tilde{X}$ introduces a quantization error or a distortion $d(X, \tilde{X})$. The average distortion over the set of input vectors X is:

$$D = \sum_{k=1}^{L} p(X \in C_k) E\left[d\left(X, \tilde{X}_k\right) / X \in C_k\right] \tag{2.14}$$

$$= \sum_{k=1}^{L} p(X \in C_k) \int_{X \varepsilon C_k} d\left(X, \tilde{X}_k\right) dx$$

Where $p(X \in C_k)$ is the probability that the vector X falls in the cell C_k and $p(X)$ is the joint pdf of n random variables. As in the case of scalar quantiziation, we can minimize D by selecting the cells.

{Ck, $1 \le k \le L$} for a given PDF p(x).

A commonly used distortion measure is the mean square error

$$d_2(x, \tilde{x}) = \frac{1}{n}(x - \tilde{x})'(x - \tilde{x}) = \frac{1}{n}\sum_{k=1}^{n}(x_k - \tilde{x}_k)^2 \tag{2.15}$$

Let us consider the partitioning of the n-dimensional space into L cells {C_k, $1 \le k \le L$} so that the average distortion is minimized over all L-level quantizers. There are two conditions for optimality. The first is that the optimal quantizer employs a nearest-neighbor selection rule, which may be expressed mathematically as:

$$Q(x) = \tilde{X}_k$$

If and only if

$$D(X, X_k) \le D(X, \tilde{X}_j) \quad ; k \ne j, \quad 1 \le j \le L$$

The second condition necessary for optimality is that each output vector $\tilde{X}_k$ be chosen to minimize the average distortion in cell C_k. In other words, $\tilde{X}_k$ is the vector in C_k that minimizes:

$$D_k = E\left[d\left(X, \tilde{X}\right) X \varepsilon C_k\right] = \int_{x\varepsilon C_k} d\left(X, \tilde{X}\right) p(X) dX \tag{2.16}$$

The vector $\tilde{X}_k$ that minimizes D_k is called the centriod of the cell [5]. Thus, these conditions for optimality can be applied to partition the n-dimensional space into cells $\{C_k, 1 \leq k \leq L\}$ when the joint pdf P(x) is known.

2.3 BASIC SIGNAL PROCESSING OPERATION IN DIGITAL COMMUNICATION

Figure 2.2 shows as the block diagram of a digital communication system. In this diagram, three basic signal processing operations are identified as: source coding, channel coding, and modulation [6]. It is assumed that the source of information is digital by nature or converted into it by design.

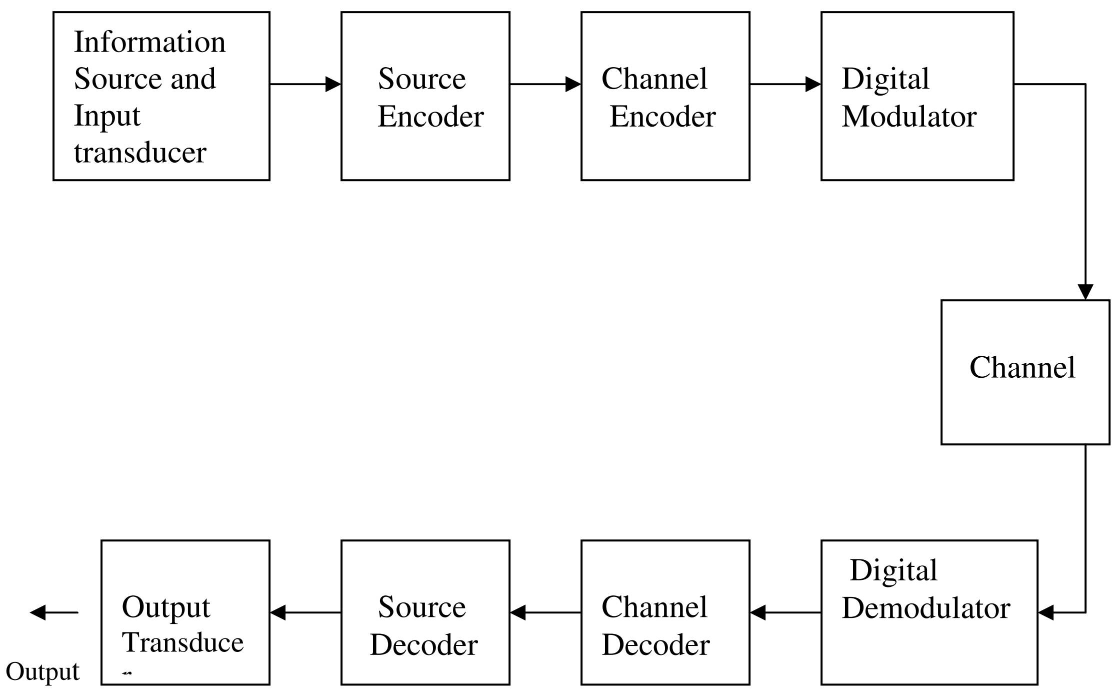


Figure 2 .2 Basic elements of digital communication system

In the source coding, the encoder maps the digital signal generated at the source output into another signal in digital form. The mapping is one to one, and the objective is to eliminate or reduce redundancy so as to provide an efficient representation of the source output. Since the source encoder mapping is one to one, the source decoder simply performs the inverse mapping and thereby delivers to the user destination a reproduction of the original digital source output. The primary benefits thus gained from the application of source coding are a reduced bandwidth requirement [6].

In channel coding, the objective is for the encoder to map the incoming digital signal into a channel input and for the decoder to map the channel output into an output digital signal in such away that the effect of the channel noise is minimized. That is, the combined role of the channel encoder and decoder is to provide for reliable communication over a noisy channel. This provision is satisfied by introducing redundancy in a prescribed fashion in the channel encoder and exploiting it in the decoder to reconstruct the original encoder input as accurately as possible. Thus, in source coding, we remove redundancy, where as in channel coding, we introduce controlled redundancy.

Clearly, we may perform source coding alone, channel coding alone, or the two together. In the latter case, naturally, the source encoding is performed first, followed by channel encoding in the transmitter as illustrated in the figure 2.2. In the receiver, we proceed in the reverse order; channel decoding is performed first, followed by source decoding. Whichever combination is used, the resulting improvement in the system performance is achieved at the cost of increased circuit complexity [5] [6].

As for modulation, it is performed with the purpose of providing for the efficient transmission of the signal over the channel. In particular, the modulator (consisting the last stage of the transmitter in fig 2.2) operates by keying shifts in the amplitude, frequency,or phase of a sinusoidal carrier wave to the channel encoder output. The digital modulation technique for so doing is referred to as amplitude–shift keying, frequency –shift keying, or phase sift keying respectively. The detector (consisting the first stage of the receiver in figure 2.2) performs demodulation (the inverse of modulation), thereby producing a signal that follows the time variation in the channel encoder output (except for the effect of noise). The combination of the modulator, channel and detector shown in figure 2.2, is called a discrete channel. It is so called since both its input and output signals are in discrete form.

Traditionally, coding and demodulation are performed as separate operation, and the introduction of redundant symbol by the channel encoder appears to imply increased transmission bandwidth. In some application, however, these two operations are performed as one function in such away that the transmission bandwidth need not be increased.

2.4 CHANNELS FOR DIGITAL COMMUNICATIONS

The details of modulation and coding used in a digital communication system depend on the characteristics of the channel and application of interest. The two channel characterstics, bandwidth and power, constitute the primary communication resource available to the designer. Other channel characteristics of particular concern are the degree to which the amplitude and phase responses of the channel are determined, whether the channel is linear or nonlinear, and how free the channel is form external interference. The following discusses these issues in the context of five specific channels: telephone channel, coaxial cables, optical fibers, and microwave radio and satellite channels [5].

A telephone channel is designed to provide voice grade communication. During the past 100 years, it has evolved into worldwide network that encompasses a variety of transmission media (open wire lines, cabels, optical fibers, microwave radio, and satellites) and a complex set of switching systems. This makes the telephone channel an excellent candidate for data communications over long distances. The channel has a band pass characteristic occupying the frequency range 300 to 3400 Hz, a high signal to noise ratio of about 30dB [5], and approximately linear response. The tuning of the channel to accommodate the transmission of voice signal results in a flat amplitude response over the pass band of the channel. However, no particular attention is given to the phase response since the ear is relatively insensitive to phase delay variations. On the other hand, data and image transmissions are strongly influenced by phase delay variation.
Accordingly, the efficient transmission of voice signals over a telephone channel requires the use of an equalizer designed to maintain a flat amplitude response and a linear phase response over the frequency band of interest. Transmission rates up to 16.8 kilobits per second (kb/s) have been achieved over telephone lines by combining the use of

sophisticated modulation techniques with adaptive equalization. The term "adaptive" refers to the fact that the coefficients of the equalizer vary in accordance with the operating conditions, such that efficient transmission is maintained.

A coaxial cable consists of a single wire conductor centered inside an outer conductor, which are insulated from each other by means of a dielectric material. The two primary advantages of coaxial cables as a transmission medium are a relatively wide bandwidth and freedom from external interference. However, their operational characteristics require the use of closely spaced repeaters. Indeed, efficient digital transmission systems using coaxial cable have been built to operator at data rate of 274 megabits per second (Mb/s), with regenerators spaced at 1 km intervals.

An optical fiber consists of a very fine inner core made of silica glass, surrounded by a concentric layer called cladding that is also made of glass. The glass in the core has a refractive index (or optical density) slightly higher than that of the glass in the cladding. A basic property of light is that when a ray of light passes from a medium of low refractive index, the ray is bent back toward the medium with higher refractive index. Accordingly, if a ray of light is launched into an optical fiber at the right oblique acceptance angle, it is continually refracted into the core by the cladding. That is, the difference between the refractive indices of the core and the cladding helps guide the propagation of the ray of light inside the core of the fiber from one end to the other. Compared to the coaxial cables, optical fibers are smaller and offer higher transmission bandwidth s and longer repeater separations.

Microwave radio, operating on a line–of–sight link, consists basically of a transmitter and a receiver that are equipped with antennas of their own. The radio channel operates as a nondispersive transmission medium, capable of highly reliable, high speed digital transmission. At other times, however, ambiguous propagation conditions develop in the channel due to metrological variation, causing sever degradation in the radio system's performance. These conditions manifest themselves in a phenomenon known as multipath fading. The term "multipath" refers to the fact that propagation between the transmitter and receiver takes place along several paths of different electrical lengths. In order to design a digital radio system to work in such an environment, it is therefore necessary that provisions must be made to overcome the effects of multipath fading.

A satellite channel consists of a satellite in a geostationary orbit. An uplink from a ground station and a downlink to another ground station. Typically, the uplink and the downlink operate at microwave frequencies. The satellite channel may be viewed as a repeater in the sky permitting communication (from one ground station to another) over long distances at high bandwidths and relatively low cost

The five channels described illustrates the divers nature of physical media that supports the transmission of digital data, this infact includes video information

2.4.1 MATHEMATICAL MODEL FOR COMMUNICATION CHANNELS

In the design of communication systems for transmitting information through physical channel, it is convenient to construct mathematical models that reflect the most important characteristics of the transmission medium. Then the mathematical model for the channel is used in the design of the channel encoder and modulator at the transmitter and the demodulater and the channel decoder at the receiver. Below, a brief description of the channel models that are frequently used to characterize many of the physical channels that we encounter in practice [5] is given.

2.4.1.1 THE ADDITIVE NOISE CHANNEL

The simplest mathematical model for a communication channel is the additive noise channel, illustrated in figure 2.3. In this model, the transmitted signal s(t) is corrupted by an additive random noise process n(t). Physically, the additive noise process may arise from electric components and amplifiers at the receiver of the communication systems, or from interference encountered during transmission (as in the case of radio signal transmission)

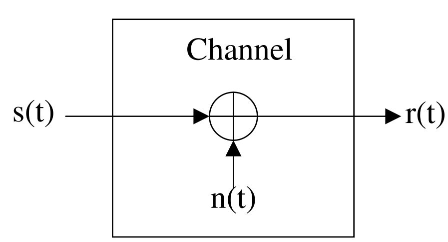


Figure 2.3 The additive noise channel

If the noise is introduced primarily by electronic components and amplifiers at the receiver, it may be characterized as thermal noise. This type of noise is characterized statistically as Gaussian noise process. Hence, the resulting mathematical model for the channel is usually called the additive Gaussian noise channel. Because this channel model applies to a broad class of physical communication channel and because of its mathematical tractability, it is the predominant channel model used in communication system analiysis and design. Channel attenuation is easily incorporated into the model. When the signal undergoes attenuation in transmission through the channel, the received signal is

$$r(t) = \alpha s(t) + n(t)$$

where α is the attenuation factor.

2.4.1.2 THE LINEAR FILTER CHANNEL

In some physical channels, such as wire line telephones channels, filters are used to ensure that the transmitted signal does not exceed specified bandwidth limitations and thus don't interfere with one another. Such channels are generally characterized mathematically as linear filter channels with additive noise, as illustrated in fig 2.4.

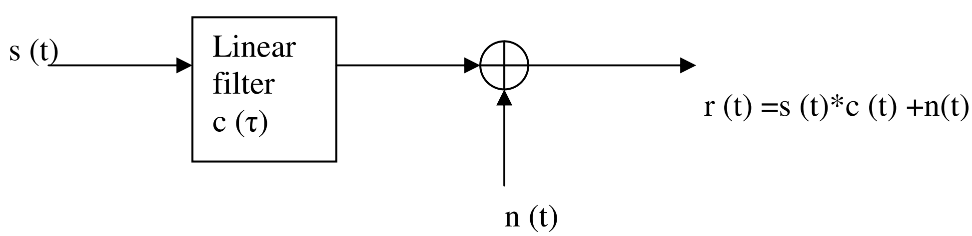


Figure 2.4. The linear filter channel with additive noise

Hence, if the channel input is the signal s(t),the channel output is the signal

$$\begin{aligned} r(t) &= s(t)*c(t) + n(t) \\ &= \int c(\tau)s(t-\tau)d\tau + n(t) \end{aligned} \tag{2.17}$$

Where c (τ) is the impulse response of the linear filter and (2.17) denotes convolution

2.4.1.3 THE LINEAR TIME VARIANT CHANNEL

Physical channels such as under water acoustic channels and ionospheric radio channels that result in time–variant multipath propagation of the transmitted signal may be characterized mathematically as time variant linear filters. Such linear filters are characterized by time–variant channel impulse response c(τ,t),where c(τ,t) is the response of the channel at time t due to an impulse applied at time τ . Thus, τ represents the "age" (elapsed time) variable. The linear time variant filter channel with additive noise is illustrated in fig.2.5 for an input signal s(t), where the channel output signal is

$$r(t) = s(t) * c(\tau ;t) + n(t)$$
$$= \int c(\tau,t)\ s(t-\tau)d\tau + n(t) \qquad (2.18)$$

A good model for multipath signal propagation through physical channels, such as ionospheric (at frequencies below 30 MHz) and mobile cellular radio channels, is a special case of (2.18) in which the time-variant impulse response has the form

$$c(\tau;t) = \sum_{k=1}^{L} a_k(t)\ \delta(\tau - \tau_k) \qquad (2.19)$$

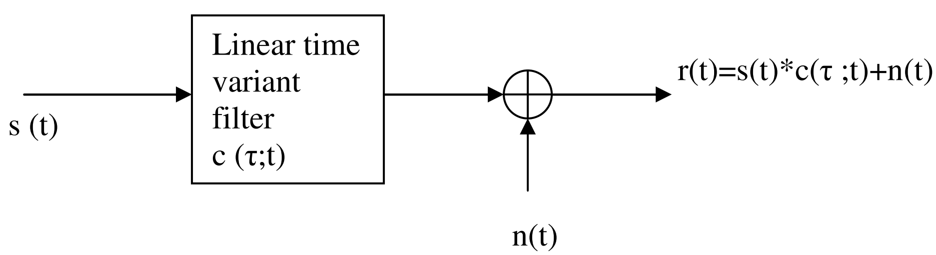


Figure 2.5 linear time variant channels

Where the $\{a_k(t)\}$ represents the possibility time –variant attenuation factor for the L multipath propagation path and $\{ \tau_k \}$ are the corresponding time delays. If (2.19) is substituted into (2.18), the received signal has the form

$$r(t) = \sum_{k=1}^{L} \int\ a_k(t)\, s(t - \tau_k)d\tau + n(t) \qquad (2.20)$$

Hence, the received signal consists of L multipath components, where each component is attenuated by $\{a_k(t)\}$ and delayed by $\{ \tau_k \}$.

The three mathematical models described above adequately characterize the great majority of the physical channels encountered in practice.

2.5 MATHEMATICAL MODELS FOR INFORMATION DISCRETE SOURCES

Any information source produces an output that is random, i.e., the source output is characterized in statistical terms. Otherwise, if the source output were known exactly, there would be no need to transmit it. The simplest type of discrete source is one that emits a sequence of letters selected from a finite alphabet. For example; a binary source emits a binary sequence of the form 10001…. where the alphabet consist of the two letters {1,0}. More generally, a discrete information source with an alphabet of L possible letters, say $\{x_1, x_2, ..., x_L\}$ has a given probability P_k of the occurrence [6]. That is

$$P_k = P(X = x_k), \quad 1 \leq k \leq L$$

Where

$$\sum_{K=1}^{L} P_k = 1$$

If the current output letter is statistically independent from all past and future outputs, then the letters in the output sequence from the source are statistically independent. A source whose output satisfies the condition of statistical independence among output letters in the sequence is said to be memoryless. Such a source is called a discrete memoryless source (DMS)

If the discrete source output is statistically dependent, the mathematical model will be based on statistical stationarity. A discrete source is said to be stationary if the joint probabilities of two sequences of length n, say $a_1, a_2, \ldots, a_n$ and $a_{1+m}, a_{2+m}, \ldots, a_{n+m}$ are identical for all $n \geq 1$ and for all shifts m. In other words, the joint probabilities for any arbitrarily length sequence of source output are invariant under a shift in the time origin [5].

An analog source has an output wave form x(t) that is sample function of a stochastic process X(t) .We assume that X(t) is a stationary stochastic process with autocorrelation function $\Phi_{xx}(\tau)$ and power spectral density $\Phi_{xx}(f)$. When X(t) is band limited stochastic process i.e ., $\Phi_{xx}(f) = 0$ for $|f| \geq W$, the sampling theorem may be used to represent X(t) as

$$X(t) = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{2W}\right) \frac{\sin[2\prod W(t - n/2W)]}{2\prod W\left(t - \frac{n}{2W}\right)} \tag{2.21}$$

Where {X(n/2W)} indicates the samples of the process X(t) taken at the sampling (nyquist) rate of f_s=2W samples/s[5]. Hence, by applying the sampling theorem, we may convert the output of an analog source into an equivalent discrete-time source. Then, the source output is characterized statistically by the joint pdf $p(x_1,x_2,\ldots\ldots\ldots,x_m)$ for all $m \geq 1$.where X_n=x(n/2w), $1 \leq n \leq m$, are random variables corresponding to samples of X(t)

2.6 MEASURE OF INFORMATION

To develop an appropriate measure of information, let us consider two discrete random variables with possible outcomes x_i, i=1, 2…n and y_i, i= 1, 2,…,m respectively. Suppose we observe some outcome Y= y_i and we wish to determine, quantitatively, the amount of information that the occurrence of the event Y= y_i provides about the event x_i,i=1, 2… n. We observe that when X and Y are statistically independent, the occurrence of the event Y= y_i provides no information about the occurrence of the event X= x_i. On the other hand, when X and Y are fully dependent such that the occurrence of Y= y_i determines the occurrence of X=x_i, the information content is simply that provided by the event X=x_i. A suitable measure that satisfies these conditions is the logarithm of the ratio of the conditional probability

$$P(X=x_i / Y= y_i) = P(x_i / y_i)$$

divided by the probability

$$P(X=x_i) = P(x_i)$$

That is the information content provided by the occurrence of the event Y= y_i about the event X=x_i is defined as

$$I(x_i; y_j) = \log \frac{p(x_i / y_j)}{p(x_i)} \tag{2.22}$$

$I(x_i; y_i)$ is called the mutual information between x_i and y_i [5][6]. The units of $I(x_i; y_i)$ are determined by the base of the logarithm, which is usually selected as ether 2 or e.When the base of the logarithm is 2, the units of $I(x_i; y_i)$ are bits, and when the base is e, the units of $I(x_i; y_i)$ are called nats

When the random variables X and Yare statistically independent $P(x_i \backslash y_i) = P(x_i)$ and, hence $I(x_i; y_i)=0$ .On the other hand, when the occurrence of the event $Y= y_i$ uniquely determines the occurrence of the event $X=x_i$, the conditional probability of the numerator of (2.22) is unity and hence

$$I(x_i; y_j) = \log \frac{1}{p(x_i)} \tag{2.23}$$

but equation (2.23) is just the information of the event $X=x_i$ .For this reason ,it is called the self information of the event $X=x_i$ and it is denoted as

$$I(x_i; y_j) = \log \frac{1}{p(x_i)} = -\log P(x_i) \tag{2.24}$$

We note that a high probability event conveys less information than a low –probability event. Infact, if there is only a single event x with the probability $p(x)=1$ then $I(x)=0$

2.7 Average mutual information and entropy

Having defined the mutual information associated with the pairs of events (x_i,y_i), which are possible outcomes of the two random variables X & Y, we can obtain the average value of the mutual information by simply weighting $I(x_i,y_i)$ by the probability of occurrence of the joint event and summing over all possible joint events. Thus, the following will be obtained:

$$I(X;Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i; y_j) I(x_i; y_j)$$

$$I(X;Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i; y_j) \log \frac{p(x_i; y_j)}{p(x_i) p(y_j)} \tag{2.25}$$

As the average mutual information between X and Y from equation (2.25) it can be seen that $I(X; Y) = 0$ when X and Y are statistically independent. An important characteristic of the average mutual information is that

$$I(X; Y) \geq 0$$

Similarly, we define the average self information denoted by H(x), as

$$H(X)=\sum_{i=1}^{n} p(x_i)I(x_i) \tag{2.26}$$

$$= -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

When X represents the alphabet of possible output letters from a source. H(x) represents the average self information per source letter, and it is called the entropy of the source. In the special case in which the letters from the source are equally probable, p(xi)=1/n for all i, and , hence:

$$H(X)=-\sum_{i=1}^{n} \frac{1}{n} \log \frac{1}{n} = \log n \tag{2.27}$$

In general, H(x) ≤ log n for any given set of source letter probabilities. In other words, the entropy of a discrete source is a maximum when the output letters are equally probable.

Considers a source that emits a sequence of statistically independent letters, where each output letter is either 0 with probability of q or 1 with probability of 1-q. The entropy of this source is:

$$H(X) \equiv H(q) = -q \log q \quad -(1-q)\log(1-q) \tag{2.28}$$

The maximum value of the entropy function occurs at q=1/2 where H (1/2) =1

2.8 Coding for discrete source

In the preceding section a measure for the information content associated with a discrete random variable X has been introduced. When X is the output of a discrete source, the entropy H(x) of the source represents the average amount of information emitted by the source. In this section, the process of encoding the output of a source will be considered, that is, the process of representing the source output by a sequence of binary digits. A measure of the efficiency of a source encoding method can be obtained by comparing the average number of binary digits per output letter from the source to the entropy H(X).

The encoding of a discrete source having a finite alphabet size may appear to be a relatively simple problem. However, this is true only when the source is memoryless. i.e., when successive symbols from the source are statistically independent and each symbol is encoded separately. The discrete memoryless source (DMS) is by far the simplest model that can be devised for a physical source. Few physical source, however, closely fit this idealized mathematical model. It's always more efficient to encode blocks of symbols instead of encoding each symbol separately. By making the block size sufficiently large, the average number of binary digits per output letter from the source can be made arbitrarily close to the entropy of the source.

2.8.1 CODING FOR DISCRETE MEMORYLESS SOURCES

For the DMS which produces an output letter or symbol every τ seconds, each symbol is selected from a finite alphabet of symbols x_i, i=1,2,…..,L occurring with probabilities $p(x_i)$, i=1,2,…..L. The entropy of the DMS in bits per source symbol is:

$$H(x) = -\sum_{i=1}^{L} p(x_i)\log_2 L \ \leq \log_2 L \qquad (2.29)$$

Where the equality holds when the symbols are equally probable. The average number of bits per symbol is H(x) and the source rate in bit/s is defined as $H(x)/\tau_s$

2.8.1.1 Fixed length code words

In a block encoding scheme that assigns a unique sets of R binary digits to each symbol, as there are L possible symbols, the number of binary digits per symbol required for unique encoding when L is a power of 2 is

$$R = \log_2 L \qquad (2.30)$$

and, when L is not a power of 2, it is

$$R = \lfloor \log_2 L \rfloor + 1$$

Where $\lfloor x \rfloor$ denotes the largest integer less than x. The code rate R in bits per symbol is now R and, since $H(x) \leq \log_2 L$, it follows that $R \geq H(x)$.

The efficiency of the encoding for the DMS is defined as the ratio H(x)/R. It will be observed that when L is a power2 and the source letters are equally probable, R=H(x). Hence, a fixed length code of R bits per symbol attains 100 percent efficiency. However, If L is not a power of 2 but the source symbols are still equally probable, R differs from H(x) by at most 1 bit per symbol. When $Log^{L}{}_{2} >> 1$, the efficiency of this encoding scheme is high. On the other hand, when L is small, the efficiency of the fixed length code can be increased by encoding a sequence of J symbols at a time. To accomplish the desired encoding, L^J unique code words are required. By using sequences of N binary digits, 2^N possible code words can be accommodated. N must be selected such that:

$$N \geq J \log_2 L$$

Hence the minimum integer value of N required is

$$N = \lfloor J \log_2 L \rfloor + 1$$

Now the average number of bits per source symbol is N/J=R, thus the inefficiency has been reduced by approximately a factor of 1/J relative to the symbol by symbol encoding described above. By making J sufficiently large, the efficiency of the encoding procedure measured by the ratio JH(x)/N, can be made as close to unity as desired.

"Let X be the ensemble of letters from a DMS with finite entropy H(x). Blocks of J symbols from the source are encoded into code words of length N from a binary alphabet, from Shannon source coding theorem. For any ε>0, the probability p_e of a block decoding failure can be made arbitrarily small if

$$R \equiv N/J \geq H(x) + \varepsilon$$

and J is sufficiently large. Conversely if,

$$R \leq H(X) - \varepsilon$$

The p_e becomes arbitrarily close to 1 as J is made sufficiently large"[5]. From this theorem it can be concluded that the average number of bits per symbol required to encode the output of a DMS with arbitrarily small probability of decoding failure is lower bounded by the source entropy H(x). On the other hand, if R<H(x), the decoding failure rate approaches 100 percent as J is arbitrarily increased.

2.8.1.2 Variable length code words

When the source symbols are not equally probable, a more efficient encoding method is to use variable length code words. In VLC, the letters that occurs more frequently is assigned short code words and those that occur infrequently are assigned long code words. We may use the probabilities of occurrence of different source letters in the selection of the code words [5][10]. The problem is to devise a method for selecting and assigning the code words to source letters such encoding is called entropy coding. The codes from entropy coding need to be uniquely and instantaneously decodable. This is true when no code word in the code is a prefix of any other code word. In general, the prefix condition requires that for a given codeword C_k of length k having elements (b_1,

$b_2, \ldots, b_k$), there is no other code word of length $L<k$ with elements ($b_1, b_2, \ldots, b_k$) for $1<L<k-1$. In other words, there is no code word of length $L<k$ that is identical to the first l binary digits of another code word of length $k>L$. This property makes the code words instantaneously decodable.

Uniquely decodable variable length codes are efficient, when the average number of bits per source letter, defined as the quantity:

$$\overline{R} = \sum_{k=1}^{L} n_k \, p(a_k) \qquad (2.13)$$

is minimized. The condition for existence of a code that satisfies the prefix condition are given by the Kraft inequality which says a necessary and sufficient condition for the existence of a binary code with code words having length $n_1 \le n_2 \le \ldots \le n_L$ that satisfy the prefix condition is

$$R = \sum_{k=1}^{L} 2^{-n_k} \le 1 \; ;$$

Where $1 \le k \le L$ and n refers to the length of codeword [5].

Huffman devised a variable length encoding algorithm, based on the source letter probabilities $p(x_i)$, $i=1,2,\ldots,L$. This algorithm is optimum in the sense that the average number of binary digit required to represent the source symbols is a minimum, subject to the constraint that the code words satisfy the prefix condition, as defined above, which allows the received sequence to be uniquely and instantaneously decodable. In the variable length encoding (Huffman), instead of encoding on a symbol-by-symbol basis, a more efficient procedure is to encode blocks of J symbols at a time.

In summary, efficient coding for a DMS may be done on a symbol-by-symbol basis using variable length code based on the Huffman algorithm. Furthermore, the efficiency of the coding procedure is increased by encoding blocks of J symbols at a time. Thus, the output of a DMS with entropy H(x) may be encoded by variable length code with an average number of bits per source letter that approaches H(x) as closely as desired.

2.9 Channel Models

The model of a digital communication system described in section 2.2 shows that the transmitter building blocks consist of the discrete-input, discrete output channel encoder followed by the modulator. The function of discrete channel encoder is to introduce, in a controlled manner, some redundancy in the binary information sequence, which can be used at the receiver to overcome the effect of noise and interference encountered in the transmission of the signal through the channel. The encoding process generally involves taking k information bits at a time and mapping each k-bit sequence into a unique n-bit sequence, called a code word. The amount of redundancy introduced by encoding of the data in this manner is measured by the ratio n/k. The reciprocal of the ratio, namely k/n, is called the code rate.

The following section describes channel models that will be useful in the design of codes. The simplest is the binary symmetric channel, which corresponds to the case with M=2

2.9.1 Binary Symmetric Channel

Consider an additive noise channel and let the modulator and demodulator/detector be included as parts of the channel. If the modulator employs binary waveforms and detector makes decision, then the composite channel, shown in the figure 2.6 has a discrete–time binary input sequence and a discrete–time binary output sequence.

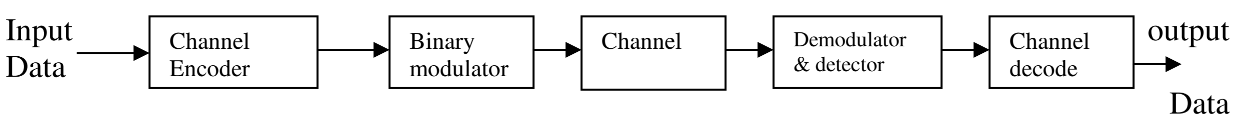
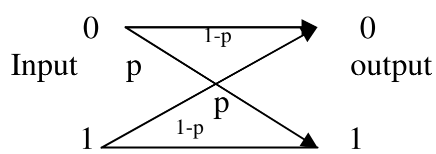


Figure 2.6 simplified block diagram for discrete–time binary input sequence and a discrete–time binary output sequence

Such a composite channel is characterized by the set X= {0, 1} of possible inputs, the set of Y={0,1} of possible outputs, and a set of conditional probabilities that relate the possible outputs to the possible inputs. If the channel noise and other disturbances causes statistically independent errors in the transmitted binary sequence with average probability P, then

$$P(Y=0/X=1)=P(Y=1/X=0)=p \tag{2.32}$$
$$P(Y=1/X=1)=P(Y=0/X=0)=1-p$$

Thus, the cascade of the binary modulator, the waveform channel, and the binary modulator will be reduced in to an equivalent discrete time channel, which is represented by the diagram shown below:



Binary Symmetric Channel

This binary input, binary output, symmetric channel is simply called a binary symmetric channel (BSC). As each output bit from the channel depends only on the corresponding input bit, the channel is memoryless.

2.9.2 Discrete memory less channel

The BSC discussed above is a special case of a more general discrete input, discrete-output channel. Suppose that the output from channel encoder are q-ary symbol, i.e., $X=\{x_0,x_1,\ldots,x_{q-1}\}$ and the output of the detector consists of Q-ary symbols, where $Q\geq m=2^q$. If the channel and the modulation are memoryless, then the input-output characteristics of the composite channel are described by a set of qQ conditional probabilities.

$$P(Y=y_i/X=x_j)=P(y_i/x_j)$$

Where i=0,1,….,Q-1 and j=0,1,2,….q-1 such a channel is called a discrete memory less channel (DMC). If the input to a DMC is a sequence of n symbols u_1, u_2,….,u_n selected from the alphabet X and the corresponding output is the sequence v_1, v_2,…..,v_n of symbols from the alphabet Y, the joint conditional probability is

$$P(Y_1=v_1,Y_2=v_2,\ldots,Y_n=v_n/X=u_1,\ldots X=u_n)=\prod_{k=1}^{n}p(Y=v_k/x=u_k) \tag{2.33}$$

The expression above is simple a mathematical statement of the memoryless condition.

In general, the condition probabilities { $P(y_i / x_j)$} that characterize a DMC can be arranged in the matrix form P=[P_{ji}], where, by definition $P_{ji} = P(y_i / x_j)$. P is called the probability transition matrix for the channel.

2.9.1.2 Discrete – input, continuous – output channel

In this case, the input to the modulator comprises symbols selected from a finite and discrete input alphabet X={$x_0, x_1, \ldots, x_{q-1}$} and the output of the detector is unquantized. Then, the input to channel decoder can assume any value on the real line, i.e. = {-∞, ∞ }. This leads the definition of a composite discrete – time memory less channel that is characterized by the discrete input X, the continuous output Y, and the set of conditional probability density functions:

$$P(y / X = x_k) \quad , \quad k=0,1,\ldots,q-1$$

The most important channel of this type is the additive while Gaussian noise (AWGN) channel, for which

$$Y=X+G$$

Where G is a zero – mean Gaussian random variable with variances σ^2 and X=Xk, k=0,1,…..,q-1. For a given X, it follows that Y is Gaussian with mean Xk & variance σ^2

$$P(y / X = x_k) = \frac{1}{\sqrt{2\prod\sigma}} e^{(y-x_k)^2 / 2\sigma^2} \qquad (2.34)$$

The condition that the channel is memory less may be expressed as

$$P(Y_1, Y_2, \ldots, Y_n / X = u_1, \ldots, X = u_n) = \prod_{i=1}^{n} P(Y_i / X_i = u_i)$$

2.9.1.3 Waveform channels

We may separate the modulator and demodulator from the physical channel, and consider a channel model in which the inputs are waveforms and the outputs are waveforms. Let us assume that such a channel has a given bandwidth W, with ideal frequency response c(f) = 1 with the band width W, and the signal at the output is corrupted by additive white Gaussian noise. Suppose that x(t) is a band limited input to such a channel and y(t) is the corresponding output. Then

$$y(t) = x(t) + n(t) \qquad (2.35)$$

Where n(t) represents a sample function of additive noise process. A suitable method for defining a set of probabilities that characterize the channel is to expand x(t), y(t) and n(t) into a complete set of orthonormal functions i.e. we express x(t), y(t) & n(t) in the form

$$y(t) = \sum_i Y_i f_i(t)$$

$$x(t) = \sum_i X_i f_i(t) \qquad (2.36)$$

$$n(t) = \sum_i n_i f_i(t)$$

Where $\{y_i\}$, $\{x_i\}$ and $\{n_i\}$ are coefficients in the corresponding expansions, such that

$$y_i = \int_0^T y(t) f_i^*(t) dt$$

$$y_i = \int_0^T [x(t) + n(t)] f_i^*(t) dt$$

$$y_i = x_i + n_i$$

It follows that

$$P(y_i / x_i) = \frac{1}{\sqrt{2\prod}\sigma_i} e^{-(yi-xi)^2 / 2\sigma^2} \qquad i=1,2\ldots \qquad (2.37)$$

Since the functions $\{f_i(t)\}$ in the expansion are orthonormal it follows that the $\{n_i\}$ are uncorrelated, again since they are Gaussian, they are also statistically independent. Hence,

$$P(y_1, y_2, \ldots, y_n / x_1, x_2, \ldots, x_n) = \prod_{i=1}^{N} P(y_i / x_i)$$

for any N. In this manner, the waveform channel is reduced to an equivalent discrete – time channel characterized by the conditional PDF given in equation (2.37).

To conclude, the choice of which channel model to use at any one time depends on our objective. If the interest is in the design and analysis of the performance of the discrete channel encoder and decoder, it is appropriate to consider a channel model in which the modulator and demodulator are a part of the composite channel. On the other hand, if the intent is to design & analyze the performance of the digital modulator and digital demodulator, it will be better to consider a channel model of the waveform channel.

2.10 Channel Capacity

Considering a DMC having an input alphabet X={$x_0,x_1,\ldots,x_{q-1}$}, an output alphabet Y={$y_0,y_1,\ldots,y_{Q-1}$} and the set of transition probabilities P(y_i/x_j), the mutual information provided about the event X=x_j by the occurrence of the event Y=y_i is log[p(y_i/x_j)/p(y)], where

$$P(y_i) \equiv p(Y = y_i) = \sum_{k=0}^{q-1} P(x_k)P(y_i / x_k)$$

Hence the average mutual information provided by the output Y about the input X is

$$I(X;Y) = \sum_{j=0}^{q-1}\sum_{i=0}^{Q-1} P(x_k)P(y_i / x_k)\log\frac{p(y_i / x_j)}{p(y_i)} \qquad (2.38)$$

The channel characteristic determine the transition probabilities P(y_i/x_j), but the probabilities of the input symbols are under the control of the discrete channel encoder. The value of I(X,Y), maximized over the set of input symbol probabilities p(x_j), is a quantity that depends only on the characteristics of the DMC through the conditional probabilities P(yi/xj). This quantity is called the capacity of the channel and denoted by C [5]. That is, the capacity of a DMC is defined as

$$C = \max_{P(x_j)} I(X;Y) \qquad (2.39)$$

$$= \max_{P(x_j)} \sum_{j=0}^{q-1}\sum_{i=0}^{Q-1} P(x_j)P(y_i / x_j)\log\frac{P(y_i / x_j)}{P(y_i)}$$

The maximization of I(X,Y) is performed under the constraints that

$$P(x_j) \geq 0$$

$$\sum_{j=0}^{q-1} P(x_j) = 1$$

The units of C are bits per input symbol into the channel (bits per channel use) when the logarithm is base 2, and nats per inputs symbol when the natural logarithm (base e) is used. If a symbol enters the channel every τ_s seconds, the channel capacity in bit/s or nat/s is C/τ_s

For discrete – time AWGN memoryless channel described by the transition probability density function by the equation (2.37), the capacity of this channel in bits per channel use is the maximum average mutual information between the discrete input $X=\{x_0,x_1,\ldots,x_{q-1}\}$ and the output $Y=\{-\infty,\infty\}$

i.e.

$$C = \max_{P(x_k)} \sum_{k=0}^{q-1} \int_{-\infty}^{\infty} P(x_k) P(y/x_k) \log \frac{p(y/x_k)}{p(y)} dy \qquad (2.40)$$

Where

$$P(y) = \sum_{k=0}^{q-1} P(y/x_k) P(x_k)$$

The necessary and sufficient condition for the set of input probabilities $\{p(x_j)\}$ to maximize I(x;y) and thus, to achieve capacity on a DMC are

$$I(x_j; Y) = C \text{ for all } j \text{ with } p(x_j) > 0$$

$$I(x_j; Y) < C \text{ for all } j \text{ with } p(x_j) = 0$$

After equating the equations above, C comes out to be

$$C = W \log\left(1 + \frac{P_{av}}{WN_0}\right)$$

This is the basic formula for the capacity of the band – limited AWGN waveform channel with a band limited and average power – limited input. The major significance of the channel capacity formula given above is that it serves as the upper limit on the transmission rate for reliable communication over noisy channel. The fundamental rate that the channel capacity plays is given by the noisy channel coding theorem due to Shannon, this theorem guarantees reliable communication, with as small an error probability as desired, if the transmission rate $R<C$. If $R>C$, it is not possible to make the probability of error tend toward zero with any code [5].

2.11 Error control coding

The most unsatisfactory feature of the channel coding theorem is its non constructive nature. The theorem only asserts the existence of good codes. But the theorem doesn't tell us how to find them. The error–control coding technique presented in the following section provide different methods of achieving reliable transmission of information over the channel. Block codes will be considered first, and it will be followed by convolutional codes [6].

2.11.1 Linear block codes

Consider an (n,k) linear block code in which the first portion of k bits is always identical to the message sequence to be transmitted. The n-k bits in the second portion are computed from the message bits in accordance with a prescribed encoding rule that determines the mathematical structure of the code. Accordingly, these n-k bits are referred to as generalized parity check bits or simply parity bits. Block codes in which the message bits are transmitted in unaltered form are called systematic codes. For application requirement both error detection and error correction, the use of systematic block codes simplifies implementation of the decoder.

Let $m_1, m_2, \ldots, m_{k-1}$, constitute a block of k arbitrary message binary bits. Thus, we have 2^k distinct message blocks. Let this sequence of message bits be applied to a linear block encoder, producing an n-bit code word whose elements are denoted by $x_0, x_1, \ldots, x_{n-1}$. Let $b_0, b_1, \ldots, b_{n-k-1}$, denote the (n-k) parity bits in the code word. For the code to possess a systematic structure, a code word is divided into two parties. One which is occupied by the message bits and the other by the parity bits. Clearly, we have the option of sending the message bits of a code word before the parity bits, or vice versa. The former option is illustrated in the figure below

$b_o, b_1, \ldots\ldots, b_{n-k-1}$	$m_0, m_1 . m_2, \ldots m_{k-1}$
Party bits	message bits

According to the representation, the (n-k) left most bits of a code word are identical to the corresponding parity bits, and the k right – most bits of the code word are identical to the corresponding message bits. We may therefore write

$$X_i = \begin{cases} b_i & ,i=0,1,\ldots,n-k-1 \\ m_{i+k-n} & i=n-k,n-k+1,\ldots,n-1 \end{cases}$$

The (n-k) parity bits are linear sums of the k message bits as shown by the generalized relation

$$b_i = P_{i,0} + P_{i,1}\, m_1 + \ldots + P_{i,k-1}\, m_{k-1}, \qquad i=0,1,\ldots,n-k-1$$

Where the coefficients are defined as follows

$$P_{ij} = \begin{cases} 1 \text{ if } b_i \text{ depends on } m_j \\ 0 \quad \text{otherwise} \end{cases}$$

These coefficients are chosen in such a way that the (n-k) equations represented in equation above are linearly independent; that is, no equation in the set can be expressed as a linear combination of the remaining ones.

There are 2^n possible code words in a binary block code of length n. From these 2^n code words, we may select $M=2^k$ code words (k<n) to form a code. Thus, a block of k information bits is mapped into a code word of length n selected from the set of $M=2^k$ code words. We refer to the resulting block code as an (n,k) code, and the ratio k/n =Rc is defined to be the rate of the code.

2.11.1.2 Some specific linear block code

In this subsection, three types of linear block codes that are frequently encountered in practice will be briefly discussed.

Hamming codes – These comprise a class of codes with the property that

$$(n,k) = (2^m-1,\ 2^m-1-m)$$

Where m is any positive integer. The parity check matrix of a Hamming code has a special property that allows us to describe rather easily. The parity check matrix of an (n,k) code has n-k rows and n columns.

For binary (n,k) Hamming code, the $n=2^m-1$ columns consist of all possible binary vectors with n-k = m elements, except the all-zero vector.

Hadamard Codes – A hadamard code is obtained by selecting as code words the rows of a Hadamard matrix. A Hadamard matrix Mn is an nxn matrix (n is an even integer) of 1s or 0s with the property that any row differs from any other row in exactly ½ n position. One row of the matrix contains all zeros. The other rows contain ½ n zeros and ½ n ones.

For n=2, the Hadamard matrix is

$$M_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

Furthermore, from M_n, we can generate the Hadamard matrix M_{2n} according to the relation

$$M_{2n} = \begin{bmatrix} M_n & M_n \\ M_n & \overline{M}_n \end{bmatrix}$$

Where $\overline{M}_n$ is the complement of M_n.

Cyclic Codes – These forms a subclass of linear block codes. Indeed, many of the important linear block codes discovered to date are either cyclic codes or closely related to cyclic codes. An advantage of cyclic codes over most other types of codes is that they are easy to encode. Furthermore, cyclic codes possess a well defined mathematical structure which has led to the development of very efficient decoding schemes for them.

A binary code is said to be cyclic code if it exhibits two fundamental properties

i. Linear property – the sum of two code words is also a code word

ii. Cyclic property – any cyclic shift of a code word is also a code word

Formulation of the cyclic property suggests that we may treat the elements of a code word of length n as the coefficients of polynomial of degree (n-1). That is, the code word with elements $x_0, x_1, \ldots, x_{n-1}$ may be represented in the form of a code word polynomial as follows

$$x(D) = x_0 + x_1 D + \ldots\ldots + x_{n-1} D^{n-1}$$

Where D is an arbitrary real variable. Naturally, for binary codes, the coefficients are 1s or Os. In general, we may describe the cyclic property in polynomial notation by stating that if x(D) is a code word polynomial, then the polynomial $D^i x(D) \bmod (D^n - 1)$ is also a code word polynomial for any cyclic shift i., where “mod” is an abbreviation for “modulo”.

2.11.2 Convolutional codes

In block coding, the encoder accepts a k–bit message block and generates an n-bits code word. Thus, code words are produced on a block by block basis. Clearly, a provision must be made in the encoder to buffer an entire message block before generating the associated codeword. There are applications, however, where the message bits come in serially rather than in large blocks, in which case the use of buffer may be undesirable. In such situation, the use of convolutional coding may be preferred method. A convolutional encoder operates on the incoming message sequence continuously in a serial manner

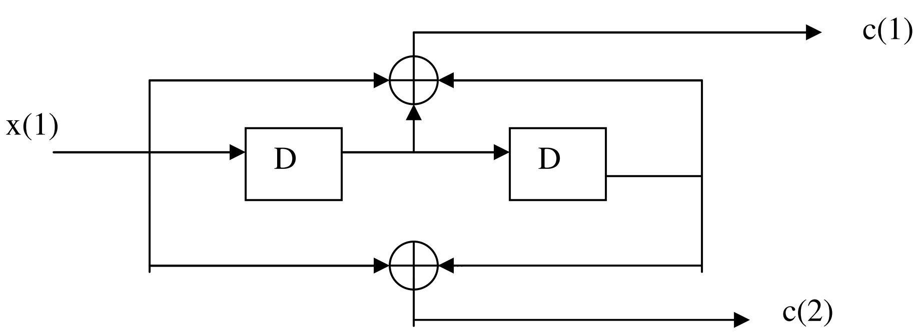


Figure 2.7: Convolutional encoder with k=1, n=2, r=1/2, m=2, and K=3 where x(i) is an input information bit stream and c(i) is an output encoded bit stream[5]

The information bits are input into shift registers and the output encoded bits are obtained by modulo-2 addition of the input information bits and the contents of the shift registers. The constraint length k for convolutional code is defined as, k=m+1, where m is the maximum number of stages (memory size) in any shift register. The shift registers store the state information of the convolutional encoder and the constraint length relates the number of bits upon which the output depends. For the convolutional encoder shown in Figure 2.7, the code rate r=1/2, the maximum memory size m=2, and the constraint length K=3.

The constraint length of a convolutional code, expressed in terms of message bits, is defined as the number of shifts over which a single message bit can influence the encoder output. In an encoder with an M–stage shift register, the memory of the encoder equals M message bits [6].

2.11.3 Error control scheme

Detecting errors is clearly of little use unless methods are available for the correction of the detected errors. Correction is thus an important aspect of data transmission. The following error control scheme methods are usually employed for error correction.

Retransmission

The most popular method of error correction is retransmission of the erroneous information. For retransmission to occur in the most expeditious manner, some form of automatic system is needed. A system which has been developed and in use is called the automatic request for repeat (ARQ), also called the positive acknowledgement/negative acknowledgement (ACK/NAK). The request for the repeat system transmits data as blocks. The parity for each block is checked upon receipt, and if no parity discrepancy is noted, a positive acknowledgement (ACK) is sent to the transmit station and the next block is transmitted. If, however, a parity error is detected, a negative acknowledgement (NAK) is made to the transmit station which will repeat the block of data. The parity check is again made and transmission continued according to the result of the parity. The value of this kind of system stems from its ability to detect errors after a small amount of data has been sent. If retransmission is needed, the redundant transmission time is held to a minimum. This is much more efficient than retransmission of the total message if only one or two data errors have occurred; however, such scheme brings delay.

Forward error – correcting codes

For transmission efficiency, error correction at the receiver without retransmission of erroneous data is naturally preferred, and a number of methods of accomplishing this are available. Codes which permit correction of errors by the receive station without retransmission are called forward error–correcting codes. The basic requirement of such codes is that sufficient redundancy be included in the transmitted data for error correction to be properly accomplished by the receiver without further input from the transmitter.

CHAPTER III
VIDEO ENCODING AND TRANSMISSION SYSTEM
3.1 INTRODUCTION

To transmit video over noisy channels, one uses both source and channel coding. According to Shannon's Separation Principle, these components can be designed independently without loss in performance [8] [13]. 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 2) an exact and complete knowledge of the statistics of the (ergodic) transmission channel. As a result of the first assumption, the Separation Principle cannot be applied without performance loss to applications with real-time constraints. This holds especially for bursty channels, which are characteristic for mobile radio transmission or the Internet. As a consequence of the second assumption, it applies only to point-to-point communications. Therefore, Joint Source-Channel Coding and Error Resilient Coding can be advantageous and have become an important research topic. By error resilience, it means that even though errors may not be detected, there are strong guarantees that their effects will not propagate.

Pragmatic approach for today's state of the art is to keep the source coder and the channel coder separate, but to optimize their parameters jointly. A key problem of this optimization is the bit allocation between source and channel coding that is also discussed in this paper. The overall performance depends on many interrelated issues, such as the distortion-rate performance and error resilience of the source codec, the error correction capability of the channel codec, and the characteristic of the channel. Because of this interaction of system components, the influence of individual parameters is difficult to understand, and the design of the overall system might become a formidable task. It is therefore desirable to develop appropriate models to study and understand the interaction and tradeoff between system parameters.

3.2. BASIC VIDEO COMMUNICATION

To give a basic over view to a video communication system, a generic functional diagram for a video transmission process in a workstation is shown in figure 3.1 [14]. The first step in the process is to analyze the supplied analog video signal. The analysis can include such operations as filtering, analog to digital conversion, computations of transform coefficients, or correlation of the pixels with prestored vector quantization patterns. Usually no compression is done during the analysis. Data is only transformed to a format that is more compressible than the original signal format.

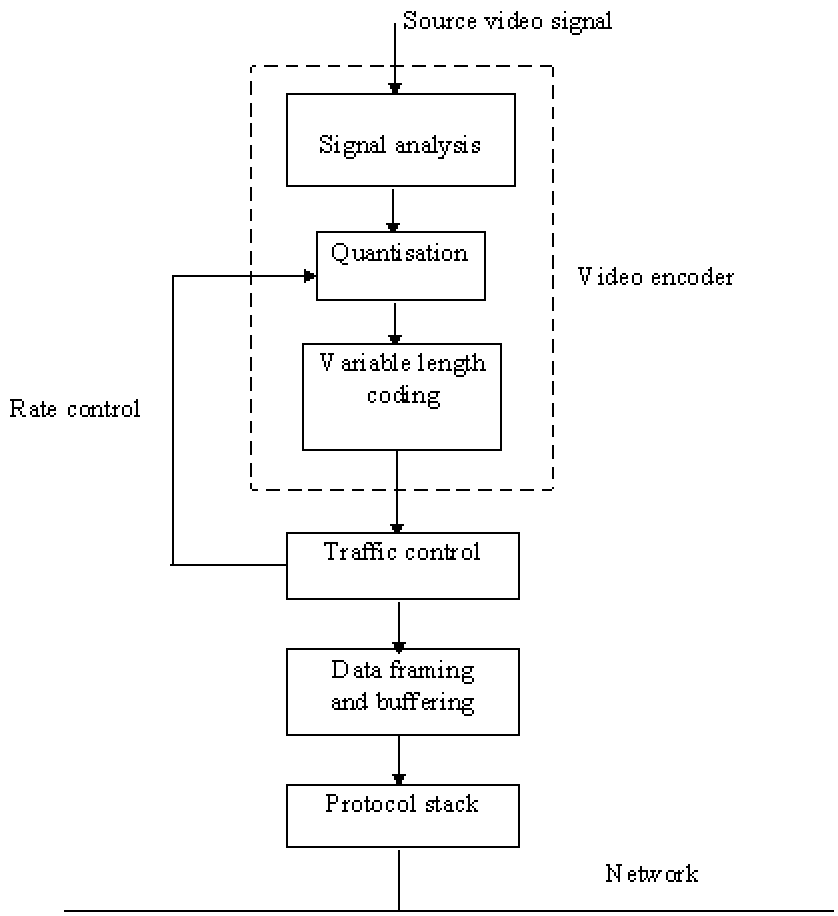


Figure3.1: A generic functional diagram for video data transmission

The second step performs quantization of the signal, in either a lossless or lossy way. In a lossy system the quantizer reduces signal accuracy in a way that is as acceptable as possible to the eye. In the variable length coding block each signal events will have a code with different number of bits. That is why it is also called entropy coding. To get compression, short codes are assigned to frequently occurring event and long codes to infrequent events.

The traffic control block follows data flow status in the communication channel, adjusts encoder parameters (Rate control in the above picture) according to the data flow status in order to adapt generated video data to the communication channel. The next block forms data packets according to the protocol employed. It also buffers the packets in order to supply a continuous and smooth data stream to the communication channel. Such system parameters as buffer size and packet length are very essential for the system performance, and therefore they must be designed very carefully. If the Real Time Protocol (RTP) is used on top of a transport protocol layer, it gives necessary information on which the parameter determination can be based on.

With reception of a video data stream, a receiver must know the received data format. For example in H.323 compliant applications it has to negotiate in the beginning of a communication session to find a suitable format for both sides. When data reception starts, the goal is that the receiver should serve video frames to its decoder in equal number of the frames, which were generated by the sender. Because of packet delay variations in the transfer path, received packets must be buffered to get required tolerance for delay variations. So called ring buffer implementation is a common solution at the receiver end [14] If we want to have more tolerance for the delay variations, more video data must be buffered and we get longer delay 'offset', accordingly. This raises an optimization problem between communication delay and the QoS. In addition to the buffering task, the receiver must also check order of packets and rearrange them if necessary, and decide what to do in case of erroneous or lost packets. RTP protocol helps greatly to implement these functions by offering required parameters such as creation time stamps in the header of the packet [14].

3.2.1. AN INTRODUCTION TO IMAGE COMPRESSION

Compressing an image is significantly different than compressing raw binary data. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. Lossless compression deals with compressing data which, when decompressed, will be an exact replica of the original data. This is the case when binary data such as executables documents etc. are compressed. They need to be exactly reproduced when decompressed. On the other hand, images (and music too) need not be reproduced 'exactly'. An approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable, that is the error between the original and the decompressed one is not noticeable to human eye.

Error Metrics

Two of the error metrics used to compare the various image compression techniques are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error. The mathematical formulae for the two are given by

$$MSE = \frac{1}{MN}\sum_{x}^{M}\sum_{y}^{N}\left[I'(x,y) - I(x,y)\right]^2 \qquad (3.1)$$

$$PSNR = 20 * \log_{10} (255 / \mathrm{sqrt}(MSE))$$

Where I(x,y) is the information of the original image, I'(x,y) is the approximated version (which is actually the decompressed image) and M,N are the dimensions of the images [2]. A lower value for MSE means lesser error, and as seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction. So, if one finds a compression scheme having a lower MSE (and a high PSNR),one can recognize that it is a better one. We'll take a close look at compressing grey scale images. The algorithms explained can be easily extended to colour images, either by processing each of the colour planes separately, or by transforming the image from RGB representation to other convenient representations like YUV in which the processing is much easier.
The usual steps involved in compressing an image are [10]

1. Specifying the Rate (bits available) and Distortion (tolerable error) parameters for the target image.

2. Dividing the image data into various classes, based on their importance.

3. Dividing the available bit budget among these classes, such that the distortion is a minimum.

4. Quantize each class separately using the bit allocation information derived in step 3.

5. Encode each class separately using an entropy coder and write to the file.

This is how 'most' image compression techniques work. But there are exceptions. One example is the Fractal Image Compression technique, where possible self similarity within the image is identified and used to reduce the amount of data required to reproduce the image. Traditionally these methods have been time consuming, but some latest methods promise to speed up the process.

Reconstructing the image from the compressed data is usually a faster process than compression [14]. The steps involved are

1. Read in the quantized data from the file, using an entropy decoder. (reverse of step 5).

2. Dequantize the data. (reverse of step 4).

3. Rebuild the image on the basis of the allocation. (reverse of step 2).

3.3. VIDEO COMPRESSION AND RELATED STANDARDS

Videoconferencing and video telephony have a wide range of applications including:

- Desktop and room –based conferencing
- Video over the internet and over telephone lines
- Surveillance, medical consultation and diagnosis at a distance
- Computer –based training and education

In each case video information is transmitted over telecommunication links, including networks, telephone lines, ISDN and radio. Video has a high “bandwidth” (i.e. many bytes of information per second) and so these applications require video compression or video coding technology to reduce the bandwidth before transmission.

As described in the preceding section, Video applications require some form of data compression to achieve reasonable precondition for storage and transmission. Digital video compression is one of the main issues in digital video coding, enabling efficient distribution and interchange of visual information [1] [10] [14].

Video codecs are devices that are used to compress and decompress as well as to encode and decode video streams. The most complex part of a codec is the compress/decompress function. Codecs can do their work by hardware but also by software with fast processors. The main goal of coding is the bit-rate reduction for storage and transmission of the video source while retaining video quality as good as possible. There are a number of international standards and also many proprietary techniques for digital video compression. The basic idea behind video compression is to remove spatial redundancy within a video frame and temporal redundancy between adjacent video frames [1].

As described above there are two main types of compression techniques, lossless and lossy. In the lossless compression a frame can be decompressed into the original exactly. The compression ratio of lossless methods is not high enough for digital video communication [14]. In the lossy compression we create compressed data that can be decompressed into images that look similar to the original (as human eye sees them) but are different in digital form.

3.3.1 Video compression techniques

The human eye is more sensitive to changes in brightness than to chromaticity changes. Therefore the image data is first divided into one luminance and two chrominance components, and the chrominance components are subsampled relative to the luminance component [1] [2][15]. After this step the usual lossy compression method used in digital video compression is based on Discrete Cosine Transform (DCT) and quantization. This technique reduces the high spatial frequency components from the image since the human viewer is more sensitive to the reconstruction errors of low frequency components. The purpose of the quantization step is to represent the DCT-coefficients with the precision that is needed to achieve the required image quality. The zig-zag step arranges the high frequency coefficients to the end of the stream and since most of them have become zero after the quantization, run length encoding (RLE) is used for further compression. As shown in the figure3.2 the upper left corner coefficient represents the mean value of the block and is encoded using the difference from the previous block (DPCM). The final step in the compression process is to minimize the entropy using Huffman or arithmetic coding. The encoded frame is often called I-frame (intra frame) because the encoding process uses no information from other frames. The block diagram of the encoding process is in Figure 3.2.

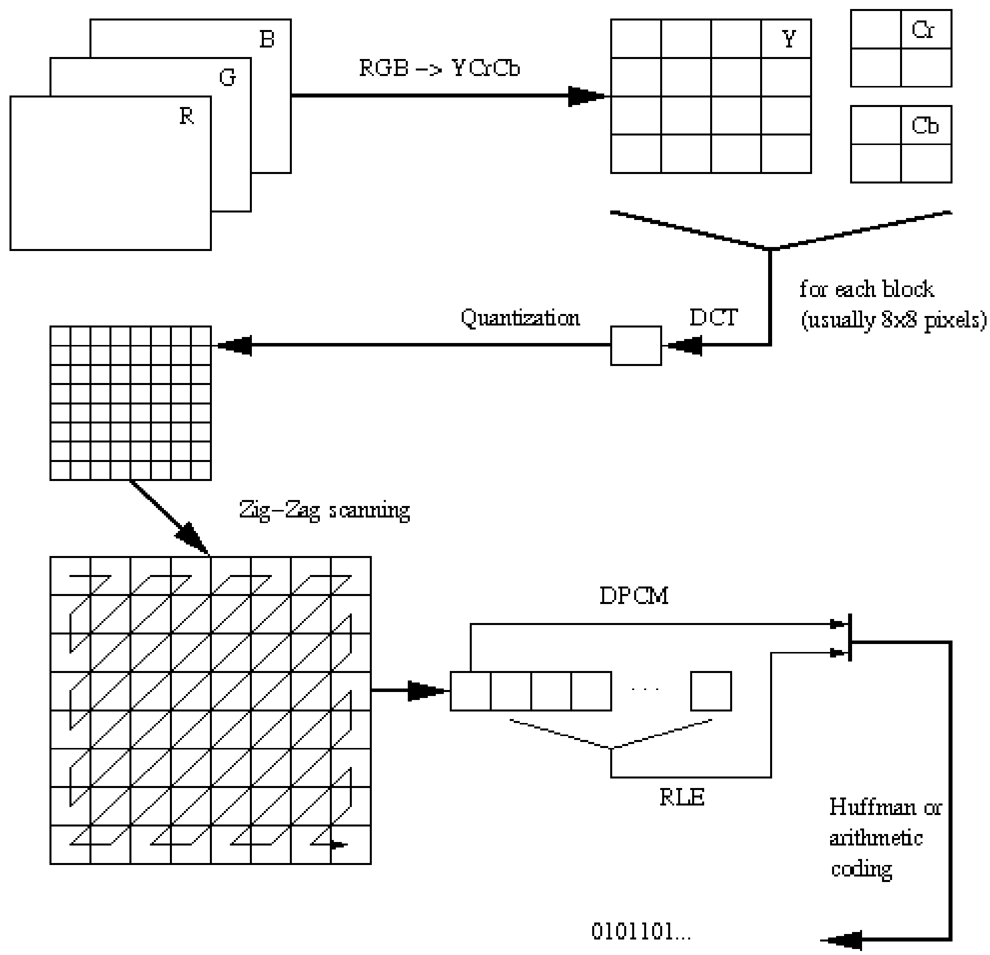


Figure3.2: Block diagram of video compression

In addition to the previous compression technique, the temporal redundancy between frames can be utilized for further compression. The basic method is to calculate the prediction error between corresponding blocks in the current and previous frames. The error values are then used in the compression process. Compressed frames generated using prediction is usually called P-frames. When using both previous and future frames as reference, the frame is called B-frame (bidirectional frame) [3].

Motion compensated prediction is an efficient tool to reduce temporal redundancy between frames. The concept of motion compensation contains the motion estimation between video frames (Figure 3.3). The motion is described by a small number of motion vectors which gives the translation of a block of pixels between frames. The motion vectors and compressed prediction errors are then transmitted.

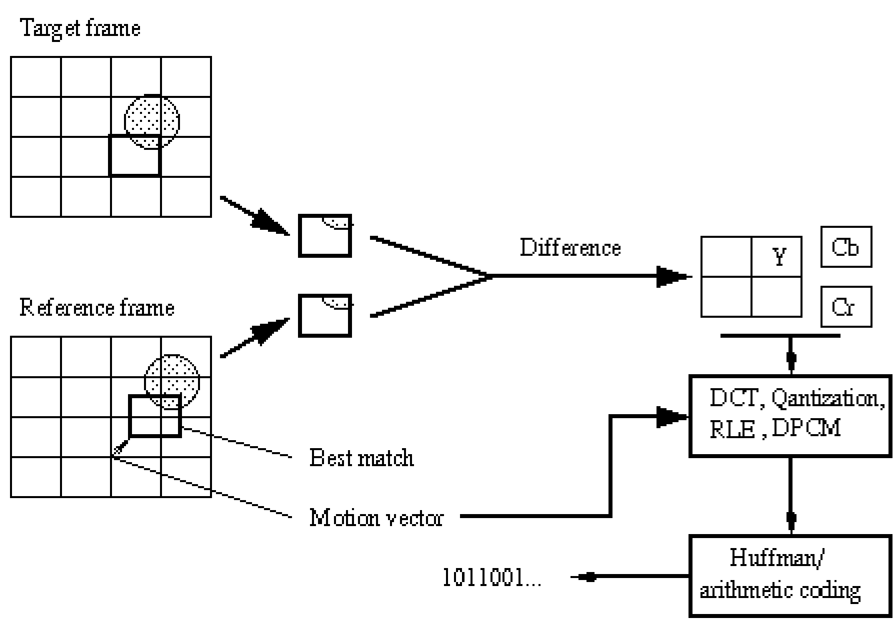


Figure3.3: Motion compensation

Newer methods no longer require this subdivision of the image and permit deep levels of compression without these artifacts, for example wavelet transformation, a powerful tool for compressing information, represents pictures, as waves that can be described mathematically in terms of frequency, energy and time. The advanced mathematics underlying wavelet transformation is very complex. The fractal compression has gained some global interest but offers so far no benefits over DCT based methods [10].

3.4. ERROR RESILIENT VIDEO CODING

A typical system is shown in the following Figure:

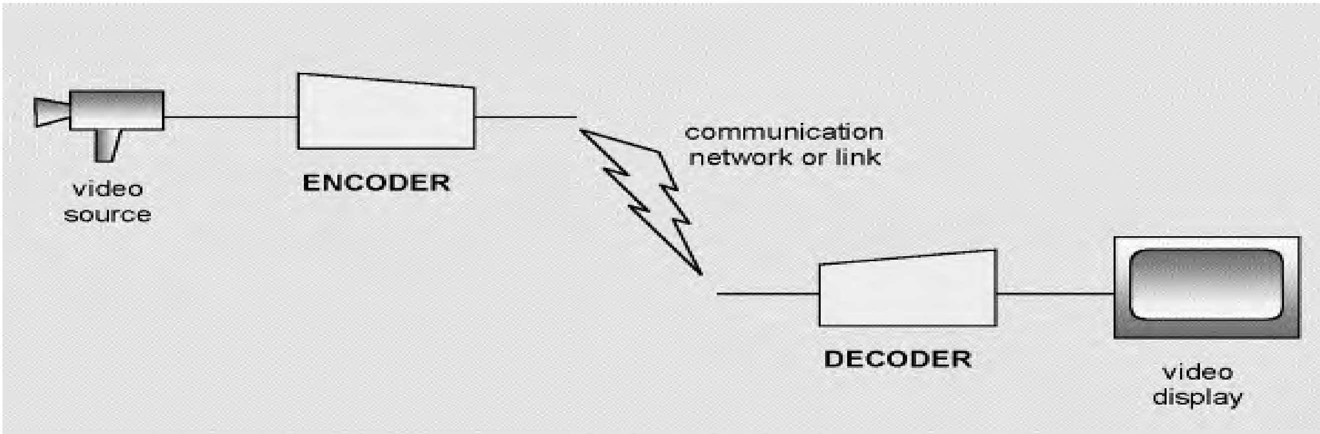


Figure 3.4: Video transmission systems

Frames of video information are captured at the source and are encoded (compressed) by a video encoder. The compressed "stream" is transmitted across a network or telecommunications link and decoded (decompressed) by a video decoder. The decoded frames can then be displayed.

A number of error resilient video coding standards exist, each of which is designed for a particular type of application: for example, JPEG for still images, MPEG2 for digital television and H.261 for ISDN video conferencing. H.263 is aimed particularly at video coding for low bit rates (typically 20-30kbps and above) [3]. The H.263 standard specifies the requirements for a video encoder and decoder. It does not describe the encoder or decoder itself: instead, it specifies the format and content of the encoded (compressed) stream. A typical encoder and decoder are described here [3][11][12].

3.4.1 VIDEO ENCODER

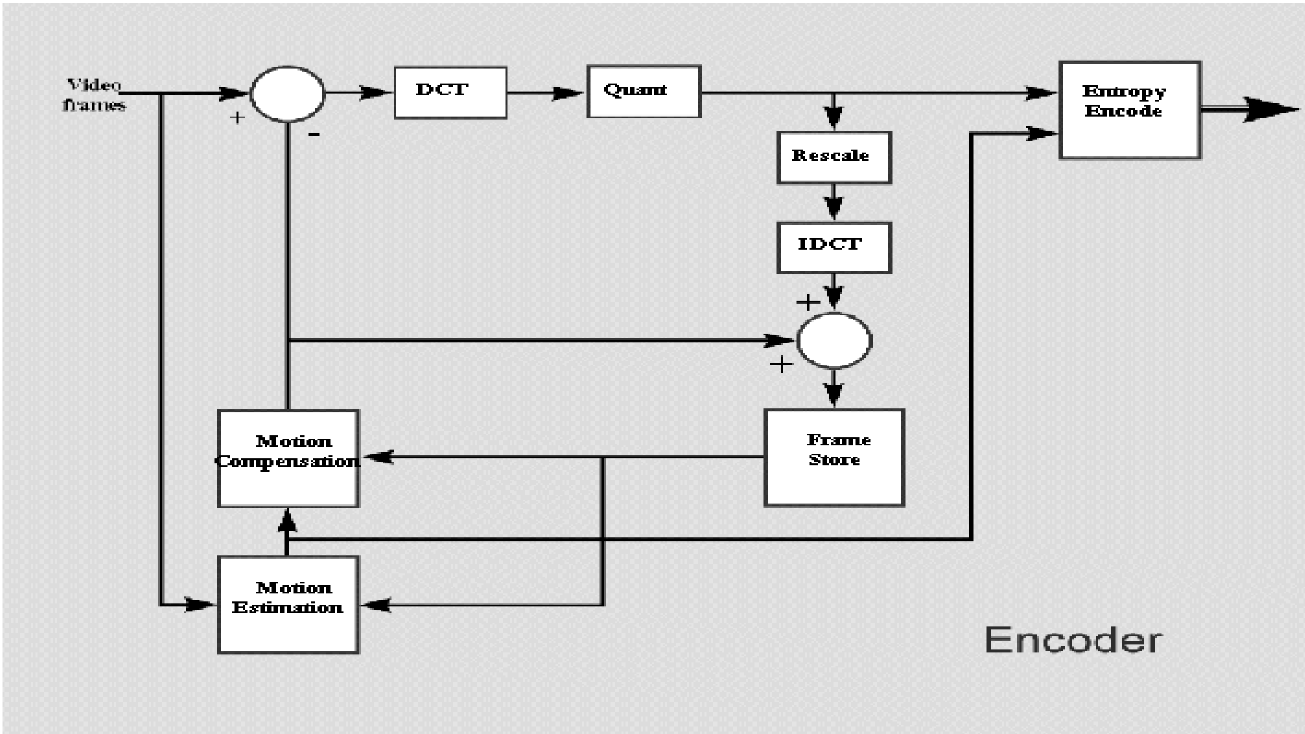


Figure 3.5: Video encoder

3.4.1.1 Motion estimation and compensation

The first step in reducing the bandwidth is to subtract the previous transmitted frame from the current frame so that only the difference or residue needs to be encoded and transmitted. This means that areas of the frame that do not change (for example the background) are not encoded. Further reduction is achieved by attempting to estimate where areas of the previous frame have moved to in the current frame (motion estimation) and compensating for this movement (motion compensation). The motion estimation module compares each 16x16 pixel block (macroblock) in the current frame with its surrounding area in the previous frame and attempts to find a match [3]. The matching area is moved into the current macroblock position by the motion compensator module. The motion compensated macroblock is subtracted from the current macroblock. If the motion estimation and compensation process is efficient, the remaining "residual" macroblock should contain only a small amount of information.

3.4.1.2 Discrete Cosine Transforms (DCT)

The DCT transforms a block of pixel values (or residual values) into a set of "spatial frequency" coefficients. This is analogous to transforming a time domain signal into a frequency domain signal using a Fast Fourier Transform. The DCT operates on a 2-dimensional block of pixels (rather than on a 1-dimensional signal) and is particularly good at "compacting" the energy in the block of values into a small number of coefficients.

3.4.1.3 Quantization

For a typical block of pixels, most of the coefficients produced by the DCT are close to zero. The quantizer module reduces the precision of each coefficient so that the near-zero coefficients are set to zero. This is done in practice by dividing each coefficient by an integer scale factor and truncating the result.

3.4.1.4 Entropy encoding

An entropy encoder (such as a Huffman encoder) codes frequently-occurring values into short binary codes and replaces infrequently-occurring values into longer binary codes. For example the entropy encoding in H.263 is based on this technique and is used to compress the quantized DCT coefficients. The result is a sequence of variable-length binary codes. These codes are combined with synchronization and control information (such as the motion "vectors" required to reconstruct the motion-compensated reference frame) to form the encoded bitstream.

3.4.1.5 Frame store

The current frame must be stored so that it can be used as a reference when the next frame is encoded. Instead of simply copying the current frame into a store, the quantized coefficients are rescaled, inverse transformed using an Inverse Discrete Cosine Transform and added to the motion-compensated reference block to create a reconstructed frame that is placed in a store (the frame store). This ensures that the contents of the frame store in the encoder are identical to the contents of the frame store in the decoder (see below). When the next frame is encoded, the motion estimator uses the contents of this frame store to determine the best matching area for motion compensation.

3.4.2 VIDEO DECODER

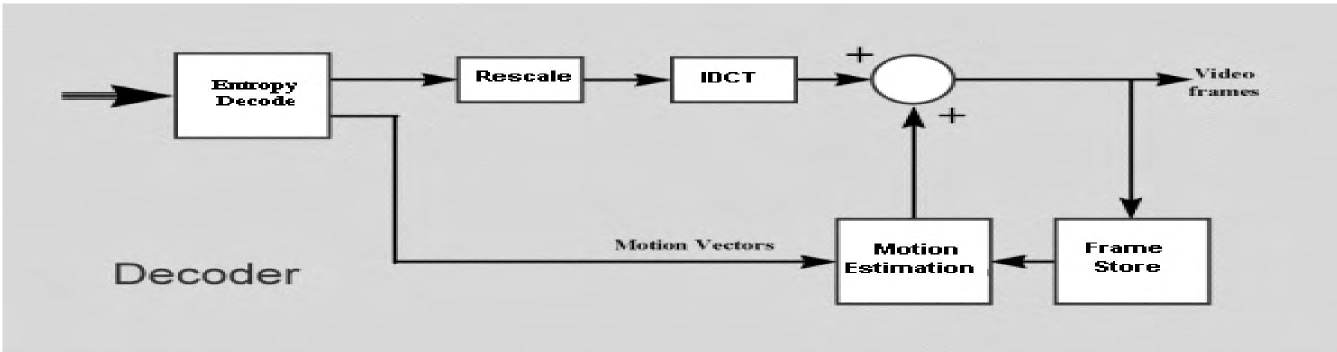


Figure 3.6: Video decoder

3.4.2.1 Entropy decoder

The variable-length codes that make up the H.263 bitstream are decoded in order to extract the coefficient values and motion vector information.

3.4.2.2 Rescale

This is the "reverse" of quantization: the coefficients are multiplied by the same scaling factor that was used in the quantizer

3.4.2.3 Inverse discrete cosine transforms

The IDCT reverses the DCT operation to create a block of samples: these (typically) correspond to the difference values that were produced by the motion compensator in the encoder.

3.4.2.4 Motion compensation

The difference values are added to a reconstructed area from the previous frame. The motion vector information is used to pick the correct area (the same reference area that was used in the encoder). The result is a reconstruction of the original frame. Note that this will not be identical to the original because of the "lossy" quantization stage, i.e. the image quality will be poorer than the original. The reconstructed frame is placed in a frame store and it is used to motion-compensate the next received frame.

3.5 VIDEO TRANSMISSION SYSTEM

This section provides an overview of the video transmission system under consideration, and introduces the most important model parameters. As can be seen from figure 3.7[8], the system consists of three parts: the video encoder, the video decoder, and the error control channel, which is defined as the combination of the channel codec and the channel.

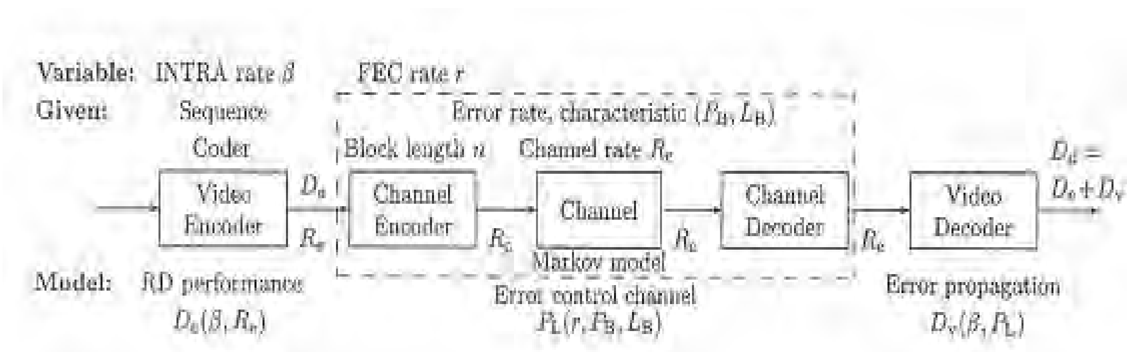


Figure 3.7: Video transmission schemes

These components are described briefly in the following paragraphs. All model parameters are summarized in Table I on page 60 for quick reference. As shown in the preceding chapters, a space-time discrete video signal is used as input to the video encoder which is characterized by its operational distortion-rate (DR) function $D_e(\beta,R_e)$; i.e., the average distortion is expressed as a function of the average bit rate R_e and INTRA rate β [8]. The common DR relationship is extended by the INTRA rate because of its significant influence on error resilience. The DR relationship, which will be discussed later, is used as the first important parameter for system optimization. After source coding, the compressed video bitstream is prepared for transmission by the channel codec. Often, this involves packetization and some form of error control; most often forward error correction (FEC) will be used which can be combined with interleaving to reduce the effect of burst errors [4] [8]. More specifically, it is assumed an (n, k) Reed–Solomon (RS) block code with a block size of symbols including $k<n$ information symbols. The second important parameter that is used for system optimization is the code rate $r=k/n$. By reducing the code rate, more channel coding redundancy is added to each codeword which improves the error correction capability of the code while reducing the throughput at the same time. After channel encoding, the RS codewords are transmitted over the channel [5][8]. A two-state Markov model is used to describe Errors at the symbol level. Channel parameters, the average symbol error rate P_B and the average burst length L_B, together with the total bit rate R_c, completely describe the channel and can be used to study the influence of burst errors verses independent symbol errors. In addition to that, the selected channel model allows calculation of the residual word error rate $P_L(r, P_{B,} L_B)$ after channel decoding from the parameters of the Markov model and the code rate. Thus, the overall performance of the error control channel, including a burst channel and an RS channel codec, can be described analytically and this will be shown in the following subsection.

Finally, the influence of residual errors on the decoded video quality has to be considered. Depending on the error resilience capabilities of the video decoder, a single lost codeword may cause severe image distortion. Fast resynchronizations of the bit-stream and error concealment are two important issues that can help to mitigate the effect of residual errors. Another important issue is interframe error propagation because errors may be visible over many consecutive frames. Hence, a model for interframe error propagation will be derived in the up coming section and this model will describe the additional distortion at the decoder D_v (β, P_L) as a function of the INTRA rate β and the residual word error rate P_L. The brief description of each system component, will enable us discuss the interactions and tradeoffs that influence the overall distortion $D_d = D_e + D_{v.}$

First consider a variation of the code rate r. Note that for a given channel bit rate R_c, the code rate controls the bit allocation between source and channel coding [8]. This has two effects on the picture quality of the video signal at the decoder output. First, a reduction of r reduces the bit rate available to the video encoder and thus increases the distortion at the encoder regardless of transmission errors. The actual D_e increase is determined by the operational DR function $D_e(\beta,R_e)$ of the video encoder. On the other hand, the residual word error rate is reduced when reducing r, determined by the properties of the error control channel according to $P_L(r, P_B, L_B)$. Finally, a reduction in P_L leads to a reduction in $D_v(\beta, P_L)$ depending on several implementation issues as discussed above. Considering the total distortion D_d at the video decoder output, these interactions of the various components make it difficult to select the optimum code rate. Basically, the characteristic of each component may have significant influence.

INTRA rate β variation can be considered as another important optimization parameter. Since INTRA coded macro blocks don't depend on the previous frame, error propagation can be reduced by increasing the number of INTRA coded micro blocks, thus the number of intra coding also reduces D_v. However, INTRA coding also reduces the coding efficiency compared to motion compensated prediction [8] [12].

3.5.1. DISTORTION MEASURE

For the evaluation of the video transmission system, it is necessary to average the distortion over the whole sequence in order to provide a single figure of merit. Even though the time averaged squared error is somewhat questionable as a measure of subjective quality, this approach is still very useful, e.g., to provide an overview for a large set of simulations. Therefore, the video quality is measured as the Mean-Squared-Error (MSE) averaged over all frames of the video sequence [1] [2]. Since PSNR is a measure more common in the video coding community, $PSNR = 10 \log_{10} (255^2/MSE)$ will be used to illustrate Simulation results. Note that the average PSNR is often computed by first computing the PSNR for each frame and averaging in time afterwards. The picture quality at the encoder and picture quality at the decoder has to be distinguished, that is using D_e to describe the over all MSE for a whole sequence. After encoding, we obtain the following corresponding PSNR value [8]

$$PSNR_e = 10 \log_{10} (255^2 / D_e) \quad (3.2)$$

Similarly, at the decoder side the PSNR value will be the following,

$$PSNR_d = 10 \log_{10} (255^2 / D_d) \quad (3.3)$$

As mentioned in the proceeding section, the overall MSE D_d is actually a superposition of two distortion types: the distortion caused by signal compression and the distortion which is caused by residual errors and interframe error propagation. Assuming that De and D_v are uncorrelated, we can calculate the overall MSE as

$$D_d = D_e + D_v \quad (3.4)$$

The above equation combines two distortion types that are likely to be perceived differently. The distortion is caused by signal compression and consists of blocking artifacts, mosquito noise, ringing, blurring, etc. The distortion introduced by transmission errors consists of severe destruction of image content and may be large and infrequent.

3.6. ANALYSIS OF THE VIDEO CODEC

In the following section, we analyze the performance of the video encoder and decoder, modeling of the distortion-rate performance of the video encoder. Then introduce an analytical model for the error propagation at the video decoder which can explain the cumulative effect of transmission errors.

3.6.1 Distortion performance of video encoder

In this section the Distortion-Rate (DR) performance of a hybrid motion compensated video encoder will be modeled. We focus on the input–output behavior of the video encoder and emphasize simplicity and usability over a complete theoretical description. On the one hand, this approach is taken because we want to describe a complete transmission system, which requires the complexity of individual components to be kept at a reasonable level. On the other hand, it is found that theoretically founded models often cannot describe experimental results very accurately due to simplistic assumptions. Although this model provides very interesting insights, it cannot describe the measured DR performance of an H.263 encoder with sufficient accuracy. Similar problems can be observed for the description of the DR performance in transform coding and DCT coding in particular [8]. Although several empirical distortion-rate models have been published, they are usually used for rate control and cannot be used to model the distortion of an entire video encoder for a given rate.

To avoid these limitations, without an increase in model complexity, we use a simple equation that relates the distortion at the encoder D_e to the relevant parameters.

For example there are two parameters with a significant impact on D_e, namely the source rate R_e that is allocated to the video encoder, and second, the percentage of INTRA coded macro blocks (INTRA rate) β that is enforced by the coding control to improve error robustness. The general idea to use empirical models to describe DR performance has also been used for rate control; however, our focus is on the description of the overall performance, i.e., the average distortion for a whole sequence given R_e and β

One drawback of this approach is that the necessary model parameters cannot be derived from commonly used signal statistics, like variance, correlation, or the power spectral density. Instead, the parameters need to be estimated by fitting the model to a subset of measured data points from the DR curve. Since we use the DR model

$$D_e = \frac{\theta}{R_e + R_0} + D_o \tag{3.5}$$

Where D_e is the distortion of encoded sequence, measured as the MSE and R_e is the output rate of the video encoder. θ, R_o and D_o are the parameters of the DR model that depend on the encoded sequence as well as on the percentage of INTRA coded macro block β. It is found that the relationship with β is approximately linear [8],

$$\begin{aligned} \theta &= \theta_p + \Delta\theta_{IP}\beta \\ R_0 &= R_{0p} + \Delta R_{oIP}\beta \\ D_o &= D_{op} + \Delta D_{oIP}\beta \end{aligned} \tag{3.6}$$

According to the above equation, the total number of model parameter is six. It is sufficient to measure the DR curves for only two different INTRA rates, intermediate values can then be obtained by linear interpolation [8].

The fitting was done by minimizing the sum of squared MSE differences between the model and the measured points. This resulted in two sets of parameters $\{\Theta, R_0, D_0\}$for each sequence. These two parameter sets together consist of six values, thus allowing us to determine Θ_p, $\Delta\Theta_{IP}$, R_{0p}, ΔR_{0IP}, D_{0p} and ΔD_{0IP} from (3.6).

The model parameters, Θ_p, $\Delta\Theta_{IP}$, R_{0p}, ΔR_{0IP}, D_{0p} and ΔD_{0IP} are used to interpolate the DR curves for other INTRA rates.

3.6.2. Theoretical framework for interframe error propagation

While motion compensated prediction yields significant gains in coding efficiency, it also introduces interframe error propagation in the case of transmission errors. Since these errors decay slowly, they are very annoying. To optimize the overall performance of video transmission systems in noisy environments, it is therefore important to consider the effect of error propagation.

It is known that two different types of errors contribute to the overall distortion at the decoder. First, the errors that are caused by signal compression at the encoder D_e and, second, errors that are caused by residual errors which cannot be corrected by the channel decoder. Since the first type of error is sufficiently described by equation (3.5), now focus will be on the second type of error and use the variable D_v to refer to it.

A simplified block diagram of a hybrid motion compensated video codec is illustrated in figure. 3.8[8], together with the relevant parameters that are introduced in the following.

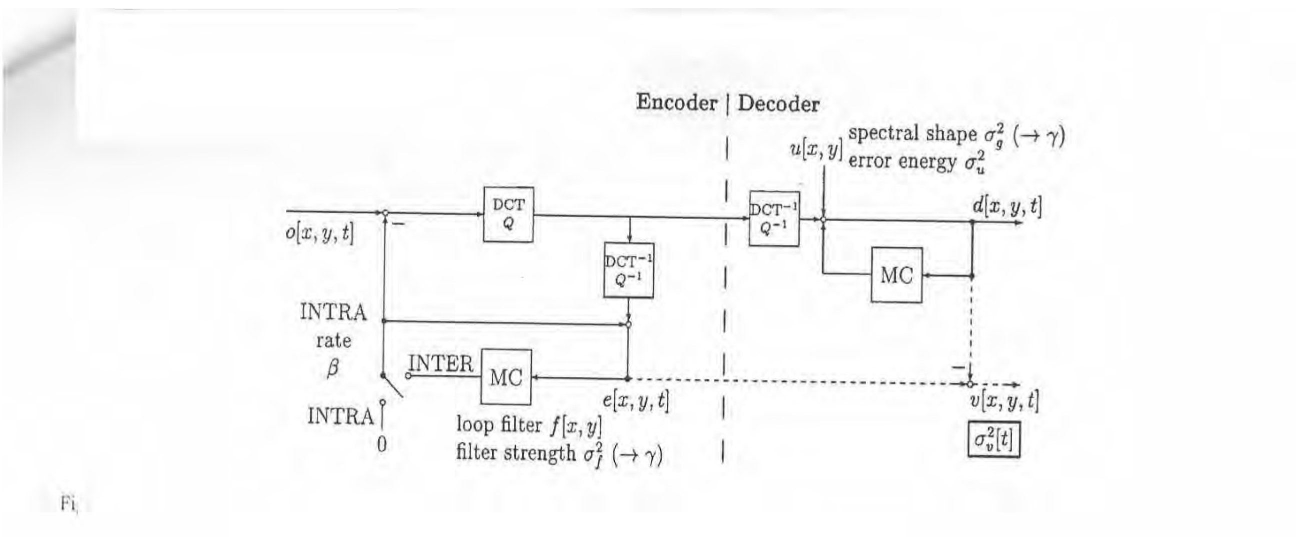


Figure 3.8: Block diagram of hybrid motion compensated video codec with transmission error[8]

Errors that are introduced by residual transmission errors are described using a stationary random process U which generates the zero-mean error signal u[x,y] . In other words, it will be assumed that, on average, the same error variance σ^2_u is introduced in each frame. Obviously, the parameter σ^2_u is directly related to the residual error rate P_L, since an increased number of lost packets will also increase the variance of introduced errors. However, it also depends on several implementation issues, like packetization, resynchronization, and error concealment, as well as on the encoded video sequence. For a given sequence, fixed packet size, and given decoder implementation, it can be shown that the error variance that is introduced can be expressed as

$$\sigma^2_u = \sigma^2_{u0} P_L \qquad (3.7)$$

This linear relation is only valid for low residual error rates, i.e.$P_L<0.1$ [4] since reasonable picture quality is very difficult to obtain for higher error rates, even when advanced error resilience techniques are employed, the given linear relation is sufficient for relevant operation conditions [8]. Note that σ^2_{u0} can be treated as a constant value that does not depend on other model parameters. It describes the sensitivity of the video decoder to an increase in error rate. If the decoder can cope well with residual errors, the value is low. For example, σ^2_{u0} can be reduced by an advanced error concealment technique [8]. Errors that are introduced at a given point in time propagate due to the recursive structure of the decoder [4] [12]. This temporal error propagation is typical for hybrid video coding that relies on motion compensated prediction in the interframe mode. It is very important to consider this effect for the design of the overall system since it has a significant influence on the sensitivity of the video decoder to residual errors. For example, even small values of σ^2_u may result in unacceptable picture quality if errors are accumulated in the decoder loop without being attenuated in some way.

Referring to figure 3.8, we are therefore interested in the accumulated error signal $\upsilon[x,y,t]$ which is the difference between the reconstructed frames at encoder and decoder, the energy of this error signal decays over time due to spatial filtering in the prediction loop and due to INTRA coding of macroblocks. More precisely, it will be derived that if the error signal $u[x,y]$ is introduced at t=0, such that $\upsilon[x,y,0] = u[x,y]$, the variance of the propagated error signal is given by

$$\sigma^2_v[t] = \sigma^2_u \frac{1-\beta t}{1+\gamma t} \qquad (3.8)$$

For $0 \le t < T$, where T is the INTRA update interval [8]. For $t \geq T$ we assume that the introduced error energy is removed completely by INTRA coded macroblocks, and hence $\sigma^2_v[t]=0$ is obtained. In other words, we assume that the implemented INTRA up-date scheme encodes each MB once in INTRA mode within an interval of encoded frames. In a practical system, some error energy might remain for $t \geq T$ due to migration by motion compensation. The relationship between T and the INTRA rate is given by [12]

$$\beta = 1/T \tag{3.9}$$

The leakage γ describes the efficiency of loop filtering to remove the introduced error. Its value depends on the strength of the loop filtering as well as on the shape of the power spectral density of the introduced error u[x,y]. If no spatial filtering is applied in the predictor $\gamma=0$, and the decay in error energy is only influenced by INTRA coding. The value of γ usually increases when more spatial filtering is applied in the predictor or when the introduced error includes high spatial frequencies that can easily be removed by the loop filter. The range of typical values is given by

$$0 < \gamma < 1. \tag{3.10}$$

Since each individual error propagates over at most T successive frames and the decoder is linear, we can derive the average distortion D_v as the superposition of error signals that are shifted in time. If we further assume that the superimposed error signals are uncorrelated from frame to frame, we can calculate D_v directly from (3.8), yielding [8]

$$D_V = \sigma_{uo} P_L \sum_{t=0}^{T-1} \frac{1-\beta t}{1+\gamma t} \tag{3.11}$$

3.6.2.1 Parameter estimation

The calculation of D_v according to (3.11) requires the knowledge of several model parameters. In the following we will discuss how to obtain these parameters. The INTRA rate β can be regarded as a control parameter that is enforced by the coding control of the video encoder, and hence is known a priori. Note that the effective average number of INTRA coded macroblocks per frame might be a little bit higher since the mode decision prefers the INTRA mode sometimes even when it is not enforced. However, there is no significant difference between the enforced and the measured INTRA rate. The parameter P_L is the residual word error rate and depends on the channel characteristic as well as on the channel codec used. The following subsection shows how P_L can be derived for a particular channel codec and channel model.

The remaining two parameters, γ and σ^2_{u0}, have to be estimated for a given video codec, packetization, and video sequence. Any two measurement points are sufficient to match γ and σ^2_{u0}. More formally, assume that two distortion values D_v are given for known values of β and P_L, i.e., $D_v^{(1)}(\beta^{(1)}, P_L^{(1)})$, and $D_v^{(2)}(\beta^{(2)}, P_L^{(2)})$. Then we can obtain the corresponding values for γ and σ^2_{u0} from (3.11) using numerical minimization. Although fitting to more measurement points can increase the robustness, two points are sufficient if selected carefully. Basically, the range of interesting INTRA rates and distortion values should be covered. Hence, we use $\beta^{(1)}=0.01$ and $\beta^{(2)}=0.33$ and select the parameters of the error control channel (code rate, symbol error rate) such that the range $10< D_v <140$ is covered [8].

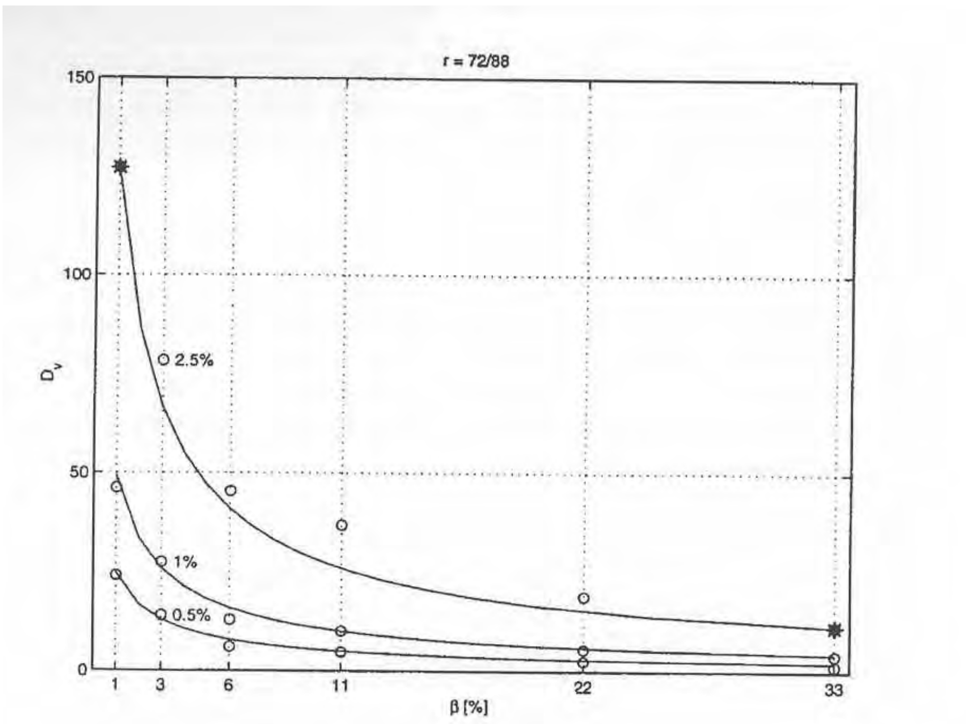


Figure 3.9

The experimental data presented in the above figure 3.9 [8] shows that equation (3.11) approximates the cumulative effect of transmission errors [8]. The measured and calculated value for D_v is plotted as a function of β for three different symbol error rates P_B, whereas γ and σ^2_{u0} are kept constant.

3.6.3 Influence of channel coding and channel parameter

The reliability of a transmission can be improved by Forward Error Correction (FEC), thus lowering the residual word error rate and the additional distortion as described by (3.11). However, in order to maintain a constant channel data rate, the available data rate for the source encoder must be reduced to

$$R_e = r\,R_c \tag{3.12}$$

Where $r \in [0,1]$ is the channel code rate. This implies that the distortion D_e (3.5) introduced by the source encoder increases. Hence, a tradeoff between source coding distortion D_e and channel induced distortion D_v results. For the optimization of the total distortion $D_d = De + Dv$, it is therefore important to understand how much reliability can actually be gained by a certain reduction in code rate.

For symbols composed of m bits, the encoder for an (k,n) RS code groups the incoming data stream into blocks of k information symbols and appends n-k parity symbols to each block. Hence, the code rate is

$$r = \frac{k}{n} \tag{3.13}$$

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 measured in bytes. For an (n, k) RS code, any error pattern resulting in less than

$$r = \frac{n-k}{2} \tag{3.14}$$

symbol errors can be corrected. Other error patterns containing more than t_c symbol errors may also be corrected with a certain probability [5][8]. The probability of undetected errors is very unlikely, especially for large n. If it still happens, it is likely to be detected by the video decoder due to syntax violations checks (which is a result of left redundancy in the source coding, e.g., sync words).

Note that the block length determines the delay introduced by the FEC scheme, because a buffer at the receiver is necessary which can hold n symbols. On the other hand, the error correction capability of the code is usually improved by increasing the block length. In fact, one result of Shannon's classical information theory is that reliable communication can always be achieved for n→∞ as long as the code rate is selected to be less than or equal to the channel capacity. However, in practical systems, the block length is limited due to delay constraints. Moreover, the complexity of the channel code does increase with the block length n and thus becomes prohibitive for practical systems at some point. Hence, an error free transmission cannot be guaranteed, i.e., there are always residual errors.

The residual word error rate P_L is the probability that a block cannot be corrected. Based on (3.14), it can be calculated as

$$P_L = \sum_{k=t_c+1}^{n} P_D(n,k) \tag{3.15}$$

where P_D is the block error density function. $P_D(n,k)$ denotes the probability of symbol errors within a block of n successively transmitted symbols [8]; e.g., for the Binary Symmetric Channel (BSC) with symbol error probability P_B ,P_D is given by the binomial distribution

$$P_{D,BSC}(n,k) = \binom{n}{k} P_B{}^K (1 - P_B)^{n-k}$$

For channels with memory, it is more complicated to calculate. We use a simple and analytically tractable 2-state Markov model with only two parameters to describe the errors on the symbol level. The two states of the model are denoted G (good) and B (bad). In state G symbols are received correctly, whereas in state B symbols are erroneous. The model is fully described by the transition probabilities P_{GB} between states G and B and P_{BG} between state B and G. Since these parameters are not intuitive, we prefer to use the error probability

$$P_B = P_r(B) = \frac{P_{GB}}{P_{GB} + P_{BG}} \tag{3.16}$$

The average burst length

$$L_B = 1 / P_{GB}$$

Which is the average number of consecutive symbol errors [5] [8]. An overview of the model includes the memoryless BSC as a special case by setting

$$L_B = 1 / (1 - P_B)$$

3.7 DECODED VIDEO QUALITY

In the following verification will be made on the derived system model by comparison to results obtained by using parameters discussed in the preceding section with practical problem. The problem at hand is to minimize the overall MSE for D_d a lossy channel by adjusting the INTRA rate (percentage of INTRA β coded macro blocks) and the FEC code rate r. First, focus will be made on the optimal intra rate.

3.7.1 Optimal INTRA Rate

In this subsection, the influence of the INTRA rate β on the decoded picture distortion D_d is studied for a fixed channel code rate r. Obviously there is a tradeoff to be considered for the selection of the INTRA rate β. On the one hand, an increased percentage of INTRA coded macro blocks helps to reduce inter-frame error propagation, and therefore reduces D_v as described by equations (3.9) and (3.10). On the other hand, a high INTRA rate increases the distortion D_e that is caused by compression at a given target bit rate. The influence of β on D_e is given by equations (3.5) and (3.6).

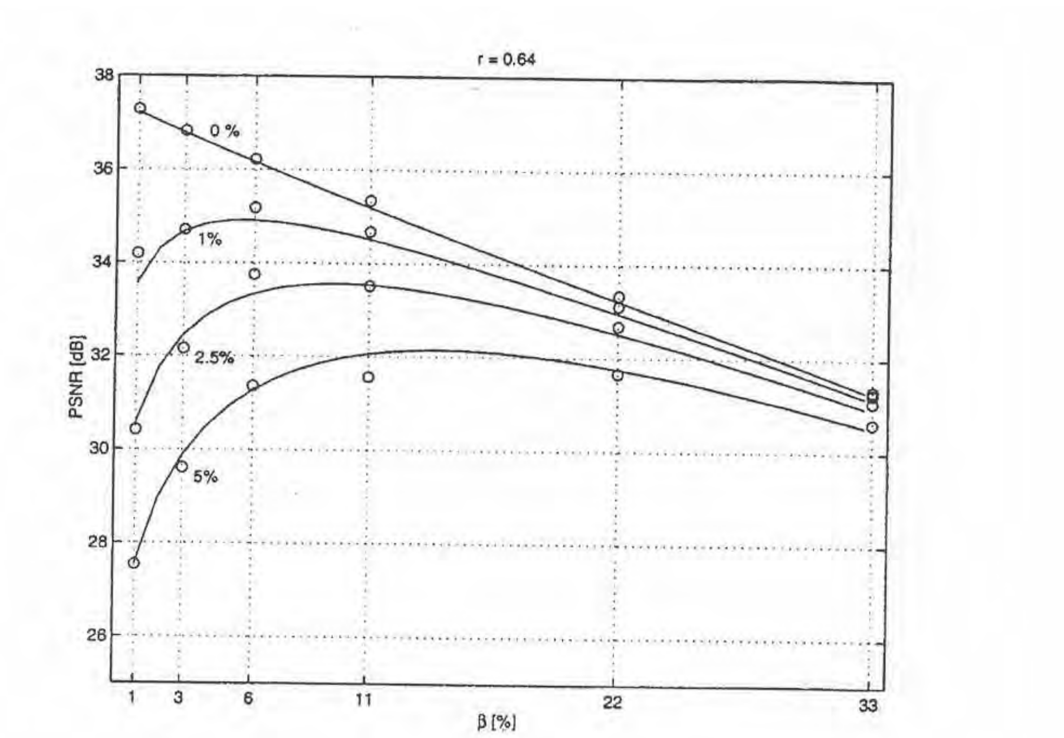


Figure 3.10

In figure 3.10[8] the video quality at the decoder PSNR is plotted against the INTRA rate for four symbol error rates P_B. It can be seen that the model gives a very good approximation of the PSNR at the decoder.

First, consider the error–free case (see P_B=0%) increasing β has quite a large influence on the PSNR at the encoder. Hence, the additional cost by coding macro blocks in INTRA mode, instead of using motion compensated prediction, is large. For sequences with more complex motion, the same increase in INTRA rate has less effect [8]. Therefore, the INTRA mode can be used more generously, and higher optimal INTRA rate result.

Note that for $\beta \rightarrow 0$, the PSNR falls rapidly as $T \rightarrow \infty$and therefore

$$D_V \approx P_L \sum_{t=0}^{\infty} \frac{1}{(1+\sigma t)} \rightarrow \infty \qquad (3.17)$$

according to (3.11). Therefore, at least a small amount of INTRA coding should always be used if transmission errors may occur. As expected, the optimal INTRA rate increases with increasing symbol error rates. However, the optimum value is also sequence specific [8].

TABLE I
SUMMERY OF MODEL PRAMTERES

	Free parameter	
β	INTRA rate (as percentage of MB)	
r	Code rate of channel code	
n	Channel coding block length	

	Sequence depended parameters	**Also depends on**
R_0	Rate offset	Encoder, β
D_0	Distortion offset	Encoder, β
θ	RD factor	Encoder, β
σ^2_{u0}	Error energy per lost packet	Concealment
γ	Leakage	
	Channel parameter	
R_c	channel rate	
P_B	symbol error rate	
L_B	average burst length	
	Dependent parameter	**Dependent on**
t_c	error correction capability of channel code	n,k
K	number of information symbols per block	r,n
R_e	video source rate	R_c,r
D_e	video source coding distortion	β,R_e, R_0 ,D_0, θ
α	Power transfer factor	γ, β
σ^2_u	error energy	σ^2_{u0}, P_L
D_v	distortion at decoder due to error propagation	σ^2_u, γ,β
P_D	block error density function	n, P_B ,L_B
P_L	residual word error rate	P_D ,n,r
D_d	overall distortion at decoder	D_e , D_v

CHAPTER IV

VIDEO CODING WITH OPTIMAL INTER/INTRA MODE

4.1 INTRODUCTION

Video signals have traditionally been transmitted over networks that provide a guaranteed Quality of Service (QoS) for the connection. Therefore, the focus of video coding has been almost exclusively concerned with compression efficiency. Some networks currently provide limited or no end-to-end QoS guarantees. Further, it is anticipated that wireless extensions to the wired backbone will result in additional QoS bottlenecks. Consequently, research on video coder design for packet-switched networks is facing major new challenges. In packet-switched networks, packets may be discarded due to buffer overflow at intermediate nodes of the network, or may be considered lost due to long queuing delays. This problem is severe. Clearly, robustness to packet loss is a crucial requirement. The problem is exacerbated in the case of predictive video coding where the prediction loop propagates errors and causes substantial, and sometimes catastrophic, deterioration of the received video signal.

A variety of techniques have been proposed to enhance the robustness of the video communication system to packet loss [12]. It is widely recognized that intra-coding is an important tool for mitigating the effects of packet loss. By switching off the inter-frame prediction loop for certain macro blocks (MB's), the reproduced blocks are no longer dependent on past frames and error propagation is stopped. However, the robustness provided by intra-coding may be costly, as it typically requires a higher bit rate than inter-coding (with prediction). Too many intra-coded MB's will significantly degrade the compression performance. Thus, the problem of switching between intra-coding and inter-coding, so as to achieve the right balance between compression efficiency and robustness, is very important

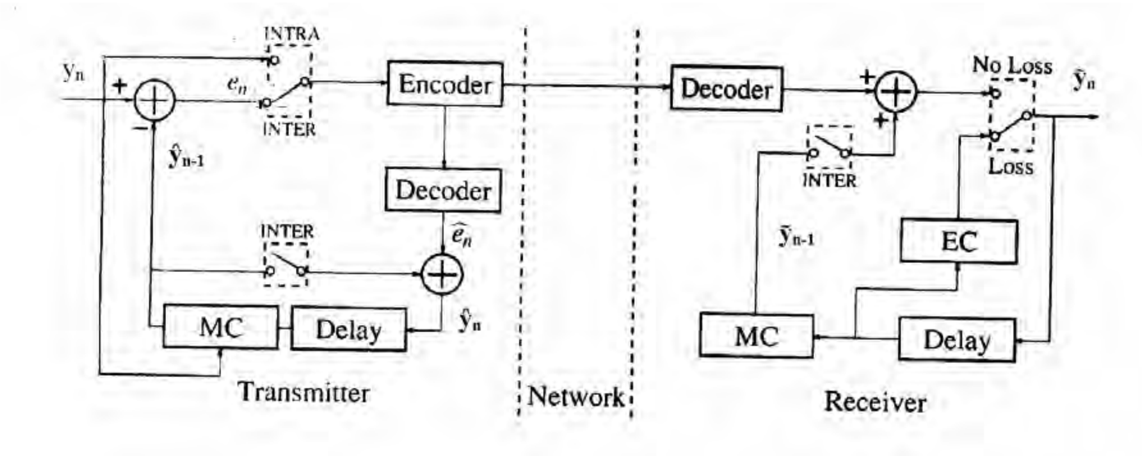


Figure 4.1: Predictive video coding with inter/intra mode switches (MC: motion compensation, EC: error concealment)

4.2 ERROR PROPAGATION AND OVERALL DISTORTION

The common video coding scheme is hybrid and employs inter-frame prediction to remove temporal redundancies, and (typically, discrete cosine) transform coding to exploit spatial redundancies. The video frame is segmented into "macroblocks" (MB's) that are sequentially encoded. Each MB may be encoded in one of two coding modes: inter-mode and intra-mode. In inter-mode, the MB is first "predicted" from the

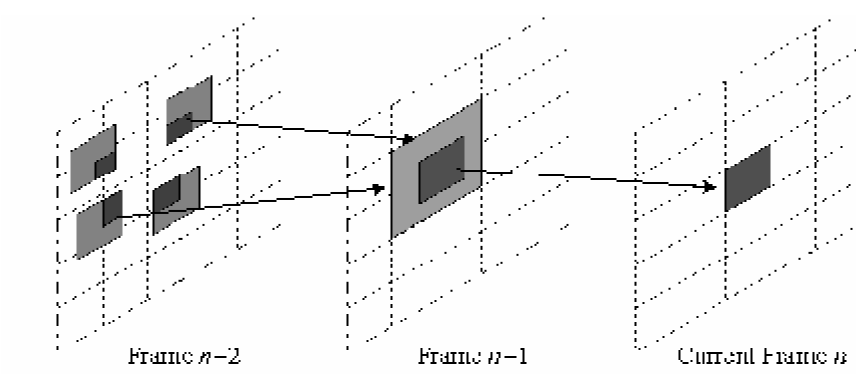


Figure 4.2: Origins of pixels in a current block, and the effect of motion Compensation on spatial and temporal error propagation.

previously decoded frame via motion compensation. Then the prediction error, or residue, is transform-coded. In intra-mode, the original MB data are transform-coded directly without recourse to prediction. Although operation in inter-mode generally achieves higher compression efficiency, it is more sensitive to channel errors as it promotes error propagation. To further illustrate this point, let us refer to the video communication system of Fig.4.1. Let a packet containing data from the current frame be lost in the channel, and let the decoder perform error concealment. Clearly, the resulting reconstruction at the decoder is different from the reconstruction at the encoder. Now, if inter-mode coding is employed to encode the next frame, errors will propagate to it via the prediction step. Whenever the motion vector is nonzero, the error propagates in both the temporal and the spatial directions. Such propagation will continue until an intra-coded block, which is independent of prior frames, is correctly received.

As shown in Fig. 4.2, the pixels in the current MB may have been motion compensated from pixels in different MB's in the previous frame, each with potentially different error propagation history. Clearly, motion compensation leads to spatial error propagation beyond MB boundaries. Hence, only by computing the estimate of the decoder reconstruction of each individual pixel we can accurately account for error propagation, and truly optimize the mode switching strategy. Further, note that for virtually all useful distortion measures, including the mean square error criterion, the distortion due to quantization and the distortion due to concealment are not additive [8][12].

4.3. RECURSIVE OPTIMAL PER-PIXEL ESTIMATE OF DECODER DISTORTION

The packet video system we consider is shown in Figure4.1 Frame n of the original video signal, denoted by y_n, is compressed and the encoder reconstruction is $\hat{y}_n$. The bits are packetized and transmitted over the network. The packets are constructed in such away that each packet can be independently decoded, and, hence, the loss of one packet does not prevent the decoding of other packets (of course, the reconstruction may still suffer from inter-frame error propagation). In this coding system, one can form a group of blocks (GOB) from all the MB's in a particular row (slice), and assume that each GOB is carried in a separate packet. In this setting, the loss rate of a pixel equals the packet loss rate p. We assume that the packet loss rate is available at the encoder. This can be either specified as part of the initial negotiations, or adaptively calculated from information provided by the transmission protocol. For example, the real time control protocol (RTCP) provides the encoder with information for calculation of packet loss rate, packet delay, and delay jitter [14]. The packets are decoded at the receiver. When a packet is lost, an error concealment technique is used for estimating the missing video segment. We denote the decoded (and possibly error-concealed) reconstruction of frame n at the receiver by $\tilde{y}_n$ .Note that the encoder does not have access to the value of $\tilde{y}_n$ and must treat it as a random variable.   The motion vector of a missing MB is estimated as the median of motion vectors of the nearest three MB's in the preceding GOB (above). If the previous GOB is lost, too, the estimated motion vector is set to zero. The pixels in the previous frame that are pointed to by the estimated motion vector are used to replace the missing pixels in the current frame.

4.4 EXPECTED DECODER DISTORTION PER PIXEL

Let y^i_n denote the original value of pixel in frame and let $\hat{y}^i_n$ denote its encoder reconstruction. The reconstructed value at the decoder, possibly after error concealment, is denoted by $\tilde{y}^i_n$. Recall that for the encoder, $\tilde{y}^i_n$ is a random variable. Using the mean square error as distortion metric, the overall expected distortion for this pixel is

$$d^i_n = E\left\{ \left(y_n^i - \tilde{y}_n^i \right)^2 \right\}$$

$$= \left(y^i_n \right)^2 - 2y^i_n E\left\{ \tilde{y}^i_n \right\} + E\left\{ \left(\tilde{y}^i_n \right)^2 \right\} \qquad (4.1)$$

We observe that the computation of d^i_n requires the first and second moments of each random variable in the sequence $\tilde{y}^i_n$.We develop recursive relationships to sequentially compute these two moments. For the recursion step, we consider two cases depending on whether the pixel belongs to an intra-coded MB or an inter-coded MB.

4.4.1 Pixel in an Intra-Coded MB

Let us first assume that the packet containing the intra-coded MB to which the pixel i belongs is received correctly. We thus have $\tilde{y}^i_n = \hat{y}^i_n$ and the probability of this event is 1-p. If the packet is lost, the decoder first checks if the previous GOB (above) has been received correctly. If the previous GOB is available, the median motion vector of the nearest three MB's is calculated and used to associate pixel i in the current frame with pixel k in the previous frame. We thus have $\tilde{y}^i_n = \tilde{y}^k_{n-1}$ and the probability of this event is p(1-p).On the other hand, if the previous GOB was lost as well, we set the motion vector estimate to zero. Thus, we have $\tilde{y}^i_n = \tilde{y}^k_{n-1}$, with probability p^2,combining the three cases; the first and second moments of $\tilde{y}^i_n$ for a pixel in an intra-coded MB are given by (intra-mode denoted by I)

$$E\left\{ \tilde{y}^i{}_n \right\}(I) = (1-P)\left(\hat{y}^i{}_n \right) + P(1-P)E\left\{ \tilde{y}^k{}_{n-1} \right\} + P^2 E\left\{ \tilde{y}^i{}_{n-1} \right\} \qquad (4.2)$$

$$E\left\{ \left(\tilde{y}^i{}_n\right)^2 \right\}(I) = (1-P)\left(\hat{y}^i{}_n \right)^2 + P(1-P)E\left\{ \left(\tilde{y}^k{}_{n-1}\right)^2 \right\} + P^2 E\left\{ \left(\tilde{y}^i{}_{n-1}\right)^2 \right\} \qquad (4.3)$$

4 .4.2 Pixel in an Inter-Coded MB

The derivation of the moments is more complex if the pixel belongs to an inter-coded MB. Let us assume that the true motion vector of the MB is such that pixel i is predicted from pixel j in the previous frame. Thus, the encoder prediction of this pixel is $\hat{y}^j_{n-1}$.The prediction error, $\hat{e}^i_{n,}$ is compressed, and we denote the quantized residue by $\hat{e}^i_{n.}$ The encoder reconstruction of this pixel, $\hat{y}^i_n$, is obtained by adding the quantized residue to the prediction. Thus

$$\hat{e}^i{}_n = \hat{y}^i{}_n - \hat{y}^j{}_{n-1} \qquad (4.4)$$

What is actually transmitted over the network is the compressed residue $\hat{e}^i_n$ and the motion vector [3][8][12]. If the current packet is correctly received, the decoder has access to both $\hat{e}^i_n$ and the motion vector. But, it must use for prediction the decoder's reconstruction of pixel j in the previous frame, $\hat{y}^j_{n-1}$, which is potentially different from the value used by the encoder. Thus the decoder reconstruction of pixel i is given by

$$\tilde{y}^i{}_n = \hat{e}^i{}_n + \tilde{y}^j{}_{n-1} \tag{4.5}$$

(Note that the value, $\tilde{y}^i_{n-1,}$ is unknown to, and therefore modeled as a random variable by, the encoder.) This explains how the error propagates in time even when subsequent frames are correctly received. The probability of the packet correctly reaching the receiver is 1-p. If the packet containing the inter-coded MB is lost, the decoder performs error concealment in a manner identical to that of an intra-coded MB. The first and second moments of $\tilde{y}^i_n$ for a pixel in an inter-coded MB are then given by (inter-mode denoted by p)

$$E\left\{\tilde{y}^i{}_n\right\}(p) = (1-p)\left(\hat{e}^i{}_n + E\left\{\tilde{y}^j{}_{n-1}\right\}\right) + p(1-p)E\left\{\tilde{y}^k{}_{n-1}\right\} + p^2 E\left\{\tilde{y}^i{}_{n-1}\right\} \tag{4.6}$$

$$E\left\{\left(\tilde{y}^i{}_n\right)^2\right\}(p) = (1-p)E\left\{\left(\hat{e}^i{}_n + \tilde{y}^j{}_{n-1}\right)^2\right\} + p(1-p)E\left\{\left(\tilde{y}^k{}_{n-1}\right)^2\right\} + p^2 E\left\{\left(\tilde{y}^i{}_{n-1}\right)^2\right\} \tag{4.7}$$

$$= (1-p)\left(\left(\hat{e}^i{}_n\right)^2 + 2\hat{e}^i{}_n E\left\{\tilde{y}^j{}_{n-1}\right\} + E\left\{\left(\tilde{y}^j{}_{n-1}\right)^2\right\}\right) + p(1-p)E\left\{\left(\tilde{y}^k{}_{n-1}\right)^2\right\} + p^2 E\left\{\left(\tilde{y}^i{}_{n-1}\right)^2\right\}$$

We reemphasize that these recursions are performed at the encoder in order to calculate the expected distortion at the decoder. The encoder can exploit this result directly in its encoding decisions, and, in particular, for mode switching. If small nonlinearities such as clipping are neglected, the above derivation is precise in the case of integer-pixel motion compensation. In the half-pixel motion compensation case, we need to take into account the bilinear interpolation performed for motion compensated prediction. The first-order moment can still be computed exactly, but the second-order moment involves computing the correlation of large matrices, and appears impractical to implement in systems of reasonable complexity. However, it is shown that the optimal estimate for this case is well approximated by the simpler recursion of integer-pixel motion and, although strictly speaking it is sub optimal, substantial gains are maintained [12].

CHAPTER V

SOURCE-CHANNEL PREDICTION IN ERROR RESILIENT VIDEO CODING

5.1 INTRODUCTION

As it is described in the preceding chapters a widely recognized critical concern with virtually all video networking application is that of efficient error resilience to adequately mitigate the impact of packet loss. Since the damage due to packet loss is greatly exacerbated by error propagation, many error resilience techniques are focused on the prediction mechanism. For example, the H.263 standard divides a picture in to non overlapping spatial regions, i.e. slices and limits spatial and temporal prediction within each slice. A video redundancy coding (VRC) suppresses the effect of temporal error propagation by partitioning input video frame into several independently predicted groups, the so called threads. When an error occur in one frame, only one thread will be affected so that the video signal can still be well reproduced with other correctly received threads [15]. Some other approaches involve multiple reference frames. Where one appropriate frame is selected out of several previously coded frames for prediction, subject to prediction accuracy and error resilience consideration. While different in many respects, these methods have one feature in common: they assume the same underlying prediction, which uses past encoder reconstructed frames for prediction, and performs motion estimation under the minimum prediction error criterion. The approach in this work is to modify conventional motion compensated prediction so as to improve or optimize the error resilience performance of the overall system

The approach here is a source-channel coding approach. Traditionally source and channel coding were separated, as motivated by Shannon's celebrated therom. However, at finite complexity and delay separate design is in general suboptimal. In fact, most encoder based error resilient methods embody this realization by attempting to account for packet loss effect during encoding [16]. The modified prediction scheme in this paper takes into account the packet loss effect and uses the expected decoder reconstruction for prediction. This is the optimal prediction in the sense of minimum end to end mean squared error (MSE) distortion as will be shown in the next section. Correspondingly motion estimation criterion will be modified, which becomes minimum expected decoder prediction error. The success of a method that minimizes decoder prediction error and distortion crucially depends on the accuracy of estimates of expected decoder quantities. So far, much research effort has been devoted to end-to-end distortion estimation. The ROPE approach, which optimally estimates the decoder reconstruction and distortion while taking in to account virtually all relevant factors including quantization, packet loss, and error concealment [12]. The optimal estimate is derived from the first and second order moments of the decoder reconstruction, which are recursively calculated for each pixel

5.2 PREDICTION BASED ON THE DECODER RECONSTRUCTION ESTIMATE

In video over internet application, the overall distortion is primarily due to two factors: quantization at the encoder and packet loss in the channel. Conventional prediction schemes use the encoder reconstruction for prediction, and hence only accounts for quantization noise. As described in the previous section, in order to improve error resilience the modified prediction scheme employs the expected decoder reconstruction for prediction, and takes into account the impact of both quantization and packet loss

Conventional prediction $$\bar{f}^i_n = \hat{f}^j_n \quad (5.1)$$

Source–channel prediction $$\bar{f}^i_n = E\{\tilde{f}^j_n\} \quad (5.2)$$

Prediction residue $$res^i_n = f^i_n - \bar{f}^i_n \quad (5.3)$$

Hence $\bar{f}^i_n$ and f^i_n denotes the original and predicted values of pixels i in frame n, respectively, while $\hat{f}^j_{n-1}$ and $\tilde{f}^j_{n-1}$ denote the encoder and decoder reconstruction values for pixel j in frame n-1,which is employed to predict pixel i in frame n(given the motion vector). The quantization residue res^j_n is the prediction error to be quantized. Note that due to packet loss the decoder reconstruction is viewed as a random variable at the decoder.

Next it will be shown that the proposed prediction in (5.2) is the optimal prediction that minimizes the expected prediction error at the decoder and, in fact, the overall end to end distortion (prior to quantization of the residual). It is necessary to emphasize that we are still making the common assumption that predictor decision are made prior to quantiztion, this means that the residue we consider is the prediction residue prior to quantization, not the quantized residue

As described in the previous chapter, the estimate of MSE end to end distortion is given by:

$$d^i{}_n = E\left\{\left(f^i{}_n - \tilde{f}^i{}_n\right)\right\} \tag{5.4}$$
$$= \left(f^i{}_n\right)^2 - 2f^i{}_n E\left\{\tilde{f}^i{}_n\right\} + E\left\{\left(\tilde{f}^i{}_n\right)^2\right\}$$

As already mentioned, ROPE is used to accurately estimate the two expected values ,namely ,the first and second moments in (5.4). Since, there is no prediction in intra macro block (MB) coding, we are only interested in inter mode MB's. For simplicity but with out implied loss of generality (a) we assume that data of one frame is carried in one packet. Hence, the pixel loss rate equals the packet loss rate (b) we assume that to conceal a lost frame it is simply replaced by the reconstructed frame. As in the preceding chapter the moments are computed recursively by

$$E\left\{\tilde{f}^i{}_n\right\} = (1-p)\left(res^i{}_n + E\left\{\tilde{f}^j{}_{n-1}\right\}\right) + pE\left\{\tilde{f}^i{}_{n-1}\right\} \tag{5.5}$$
$$E\left\{\left(\tilde{f}^i{}_n\right)^2\right\} = (1-p)E\left\{\left(res^i{}_n + \tilde{f}^j{}_{n-1}\right)^2\right\} + pE\left\{\left(\tilde{f}^j{}_{n-1}\right)^2\right\}$$

Substituting (5.5) and (5.3) into (5.4), we can get

$$d^i{}_n = (1-p)\left(\bar{f}^i{}_n - E\left\{\tilde{f}^j{}_{n-1}\right\}\right)^2 + (1-p)\left(E\left\{\left(\tilde{f}^j{}_{n-1}\right)^2\right\} - E^2\left\{\tilde{f}^j{}_{n-1}\right\}\right) + pE\left\{\left(\tilde{f}^j{}_{n-1}\right)^2\right\} \tag{5.6}$$

It is easy to see that for a given motion vector (j fixed) distortion of (5.6) is minimized by (5.2), which proves the optimality of the proposed source–channel predictor

5.3 MODIFIED MOTION ESTIMATION CRITERION

In the conventional scheme, the motion estimation criterion is to minimize the encoder prediction error

$$\min_{mv} \sum_{i \varepsilon MB} \left(f^i{}_n - \bar{f}^i{}_n\right)^2 = \min_{mv} \sum_{i \varepsilon MB} \left(f^i{}_n - \hat{f}^{i+mv}{}_n\right)^2$$

here, mv denotes the motion vector assigned to the specific MB. Obviously, as we replace the conventional predictor of (5.1) with source–channel predictor proposed in (5.2), we must reconsider and accordingly modify the motion estimation criterion of(5.7).

It turns out that there is more subtlety to this problem than may initially be expected. One may consider two different motion estimation criteria:

Criterion I: $$\min_{mv} \sum_{i \varepsilon MB} \left(f^{i}_{n} - E\left\{ \hat{f}^{i+mv}_{n-1} \right\} \right)^2 \qquad (5.8)$$

Criterion II: $$\min_{mv} \sum_{i \varepsilon MB} E\left\{ \left(f^{i}_{n} - \hat{f}^{i+mv}_{n-1} \right)^2 \right\} \qquad (5.9)$$

Note that criterion I (5.8) is a natural choice for criterion as it considers the difference between the original pixels and the actual prediction employed at the encoder (which was appropriately modified to reflect the expected reconstruction at the decoder). However, criterion II (5.9) computes an expectation over the actual prediction error at the decoder and explicitly accounts for the fact that the decoder predictor is random. In other words, although the best predictor to use is given in (5.2) and indeed appears in (5.8), the best motion vector is the one that minimizes (5.9), which is exactly the expected square prediction error at the decoder. Cleary, criterion II is theoretically superior to criterion I, and is hence adopted in the proposed scheme

To further illustrate the difference between the two criteria, equation (5.9) will be re-expressed in terms of distortions:

$$\min_{mv} \sum_{i \varepsilon MB} E\left\{ \left(f^{i}_{n} - \hat{f}^{i+mv}_{n-1} \right)^2 \right\}$$

$$= \min_{mv} \sum_{i \varepsilon MB} \left[\left(f^{i}_{n} - E\left\{ \tilde{f}^{i+mv}_{n-1} \right\} \right)^2 + \left(E\left\{ \left(f^{i+mv}_{n-1} \right)^2 \right\} - E^2\left\{ f^{i+mv}_{n-1} \right\} \right) \right] \qquad (5.10)$$

$$= \min_{mv} \left[(1-p) D_R + D_D \right]$$

Where

$$D_R = \sum_{i \varepsilon MB} \left(\bar{f}^{i}{}_{n} - \tilde{f}^{i}{}_{n} \right)^2$$

$$D_D = \sum_{i \varepsilon MB} E\left\{ \left(f^{i}{}_{n} - \tilde{f}^{i}{}_{n} \right)^2 \right\}$$

Bearing the R-D perspective in mind, the squared prediction residue at the decoder is denoted by D_R as it directly impacts bit rate, and we denote the end to end distortion by D_D.We can see from(5.10) that Criterion I only considers D_R while Criterion II considers the property weighted impacts of D_D and D_R

5.4 SIMULATION RESULT

The simulation system is based on the H.263 codec [3]. A sequence is encoded into an H.263 bit stream given the packet loss rate and total bit rate. The bit stream then undergoes a packet loss pattern that is randomly generated with the prescribed packet loss rate. The system performance is measured by the average luminance PSNR.

In the following figures, the conventional encoder based prediction is labeled as 'conventional '. The modified source-channel prediction scheme is denoted by [criterion II of (5.9) for motion estimation] 'C-II'. Source–channel prediction whose motion estimation minimizes criterion I of (5.9), ienoted by "C-I" is also included.

In the simulation, test is made for three methods under three different conditions, namely no intra updating, periodic intra updating and R-D optimized intra updating. In the no intra updating case intra coding is disabled that is all the MB's are coded in the inter mode. Periodic intra up dating means that an MB is coded in intra mode once per 1/p frame, where p is the packet loss rate. The R-D optimized intra updating performs the coding mode selection for each MB with the R-D criterion. The objective result obtained is depicted in figures (5.1)(5.2)(5.3), from these figures, it is evident that in all the scenarios and at all packet loss rate the modified method, which is motion estimation based on criterion II of equation (5.9) offers the best performance.

The subjective result is depicted in the figure (5.4). Originally the size of the image was 297KB; the H.263 codec compresses it to 6KB and gives decoded image of similar size as that of the original one. Matlab program is used to get subjective difference between the original and the decoded one, the result is shown in the figure (5.5).

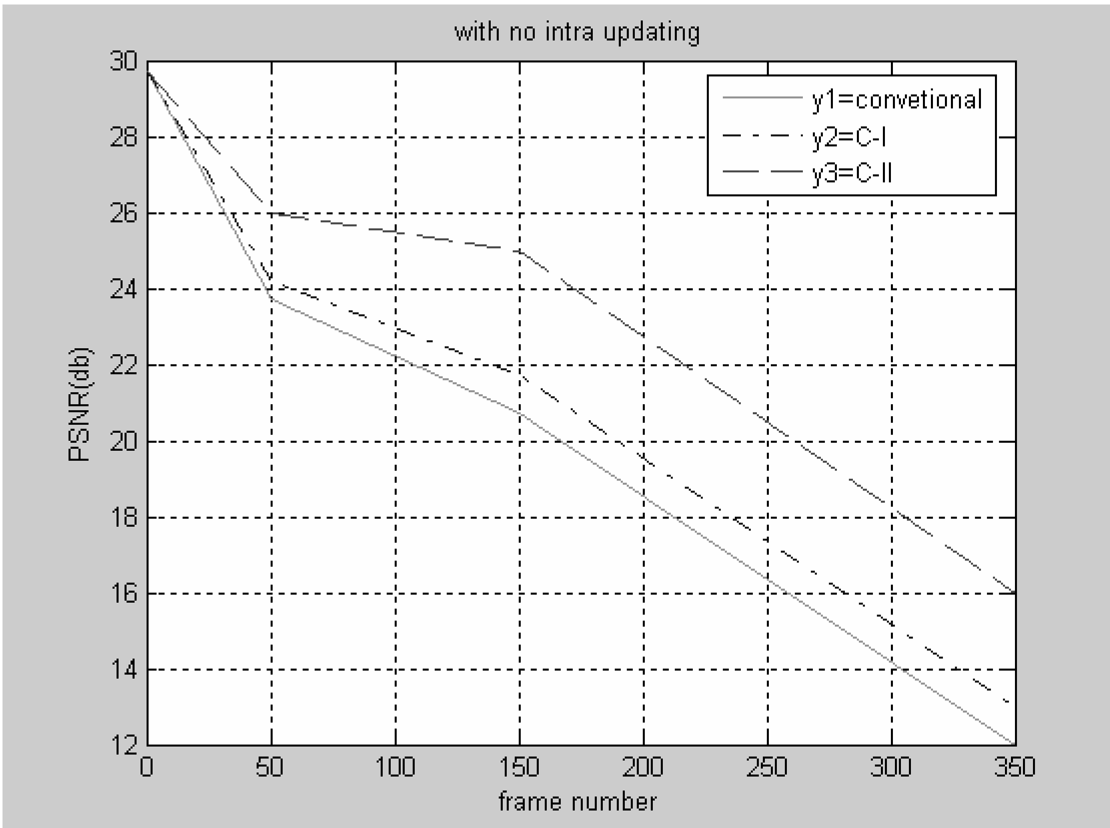


Figure5.1 : PSNR performance comparison with no intra updating

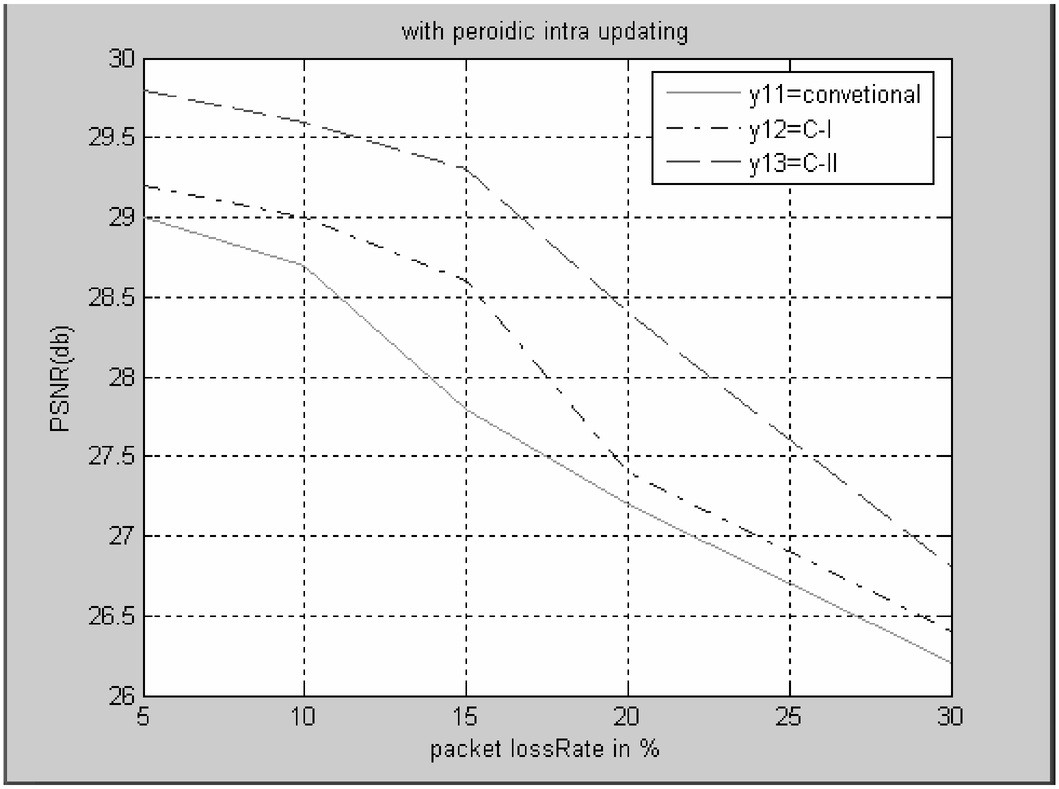


Figure5.2 PSNR performance comparison with periodic intra updating

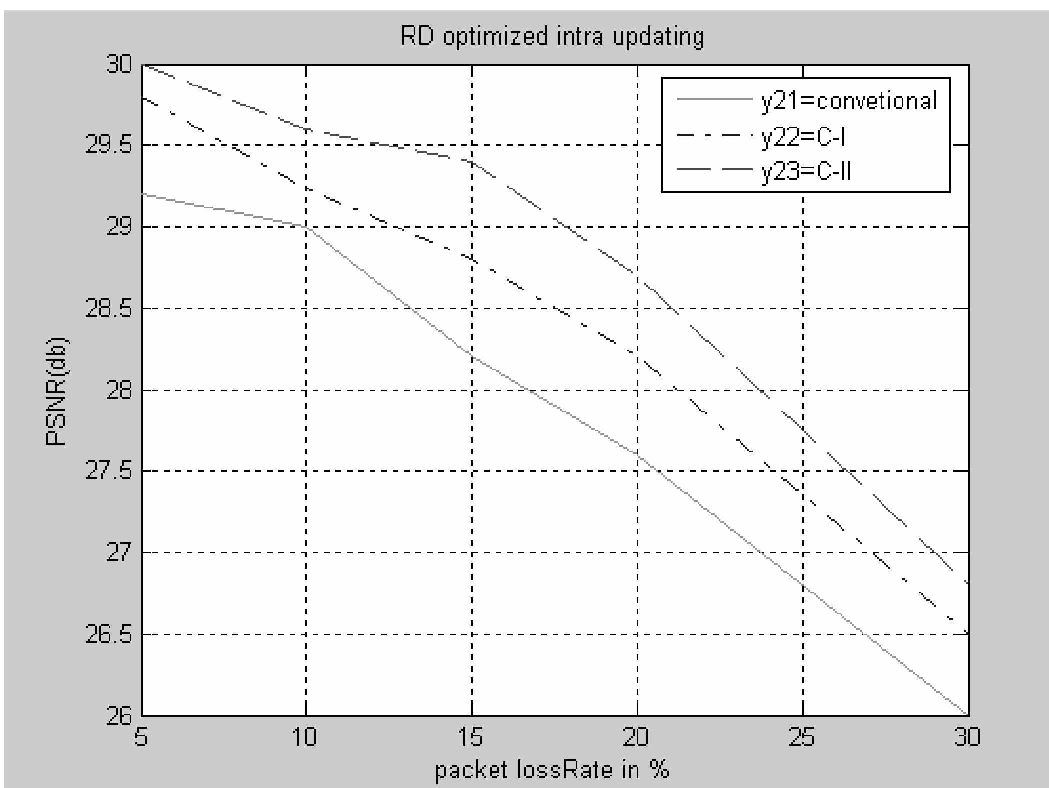
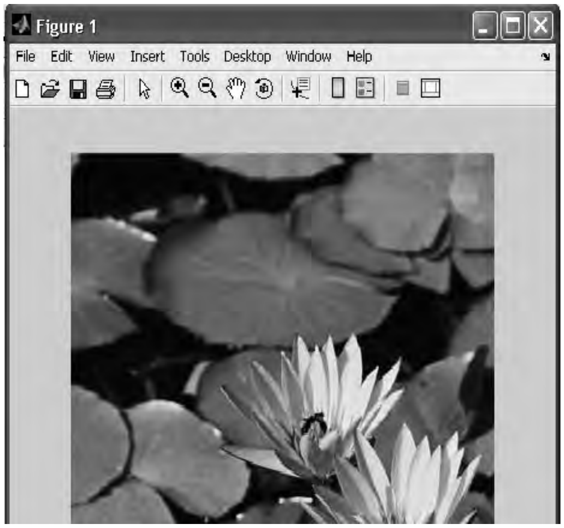
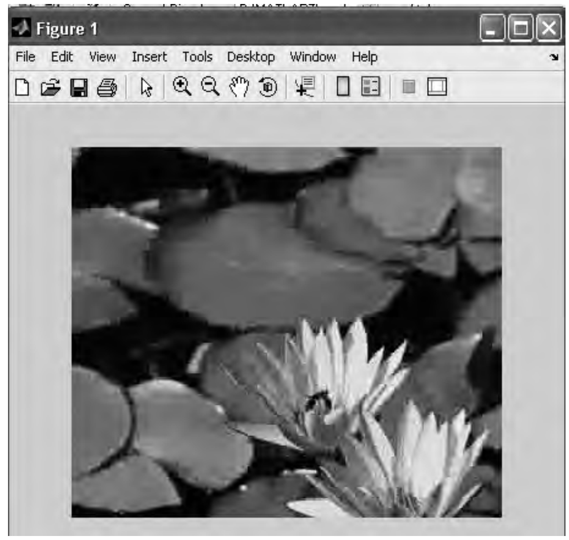


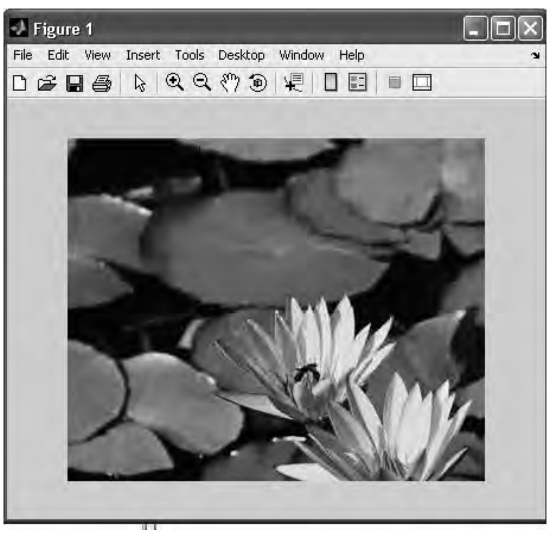
Figure5.3 : PSNR performance comparison with R-D intra updating



Original image



Decoded image with no intra updating



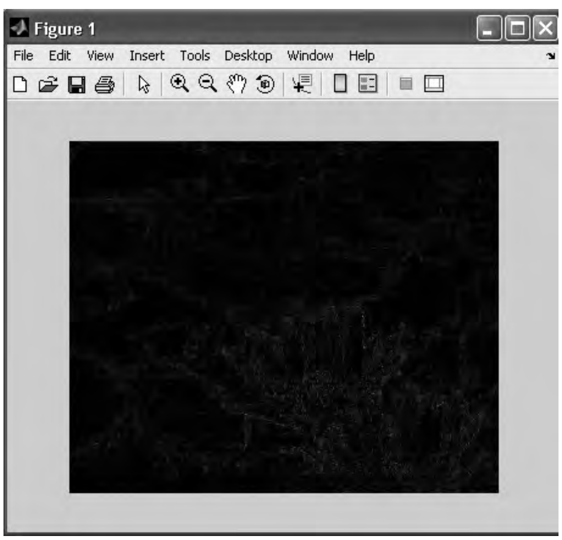
Decoded image with periodic intra updating

Decoded image with RD intra updating

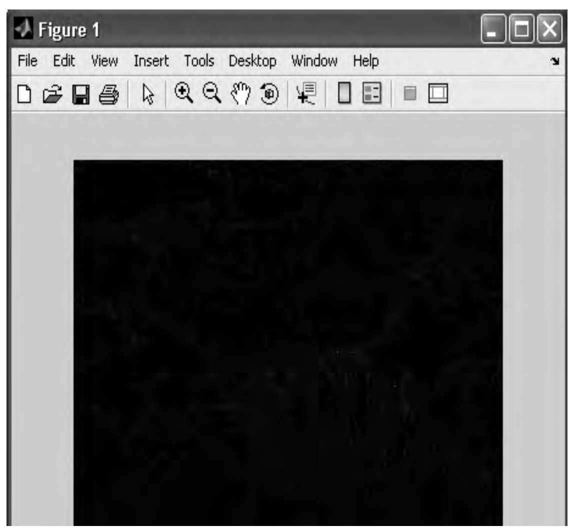
Figure 5.4 subjective results with parameter shown below

```
Input Bitmaps: kib<xxx>.bmp
Output Bitmaps: test<xxx>.bmp
Stream File: out.s
Resolution: CIF (352x288)
Frame Rate: 10
Bit Rate (Kbps): 128
Selected configuration:
Selected options:
Input BMPs    : kibxxxxx.bmp
Output BMPs   : testxxxxx.bmp
Output Stream: out.s
Standard      : H263
Resolution    : CIF
Frame rate    : 10
Bit rate      : 128
Mode          :  LS
Encoding P Frame 0...kib00000.bmp.....Success
OK (5938 Bytes)
Encoding P Frame 1.......Can't open file: kib00001.bmp
NO MORE IMAGES!
KBits/sec = 463.906250
Time (ms) 16 for 1 frames => 62.500000 frames per second
Decoding...P Frame 0
OUT OF DATA!
Time (ms) 0 for 1 frames => 1.#INF00 frames per second

H:\h26xCodec\rmd_stdcall>
```



The difference between the original and
Image decoded with periodic
intra updating from figure 5.4



difference between the original and
image decoded with RD
intra up dating from fig 5.4

Figure 5.5

5.5 Conclusions

The thesis work primarily lies in further enhancement of error resilience via fundamental modification of the conventional prediction structure. Rather than predict based on the encoder-reconstructed signal, the thesis work proposes to predict based on the expected decoder reconstruction. Such an approach hinges on accurate estimation of decoder quantities and hence estimation is built on the ROPE approach for recursive decoder reconstruction estimation. In spite of the loss in source coding, source-channel prediction achieves better over all R-D trade off than the conventional scheme. Moreover, in conjunction with the proposed source–channel predictor, it is necessary to modify the motion estimation criterion. This subtle point is shown to be of considerably significance to the performance of the system

References

[1] Janssen, ruud. ”Computational image quality” 2001

[2] Lee d.davisson and Robert m.gray”data compression” Stanford university

[3] 4i2i documentation http://www.4i2i.com/h263_video_codec.htm

[4] Stefan Goss, wilhelmVogt “Transmission of still and moving images over narrow channels” Texas instruments 1994

[5] Proakis John.G “Digital Communication” third edition 1995

[6] Simon Haykin “Digital communications” McMaster university 2000

[7] Ziemer, Roger E.”Digital communication and spread spectrum system’1985

[8] IEEE, Michael link and Bernd Girod “analysis of video transmission over lossy channels” *June 2000*

[9] Gilbert Held “Dictionary of communication technology ” 3rd edition

[10] Dr.Dobb’s Journal”Arithimetic coding +statistical modeling =data compression” *Feb, 1991*

[11] ITU-t recommendation H.263 “video coding for low bit rate communication”

[12] R.zhang and K.Rose ,”video coding with optimal inter/intra –mode switching packet loss resilience “ *IEEE J.select vol.18 no.6 June 2000 pp.966-976*

[13] T.M cover and J.A Thomas, elementary information theory network wiley 1991

[14] Transferring real-time video on the internet *http://www.tml.tkk.fi/Opinnot/Tik-110.551/1997/iwsem.html*

[15] S.wenger ,” video redundancy coding in H.263” *proc AVSP,Aberden u.k sept1997*

[16] Y.wang,Q.zhu,”error control and concealment for video communication: a review ”*proc, of IEEE, vol.86 . no 5 , may 1997 ,pp 974-997*

Appendix I

```
/* calculate the PSNR difference between two image files */

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <sys/types.h>
#include <sys/stat.h>

int main(int argc, char *argv[])
{
  /* declaring variables */
  FILE *file1, *file2;

  register long i;
  long file_size;

  char *buf1, *buf2;

  struct stat stat1;
  struct stat stat2;

  double x, error, psnr;

  /* parse the command line */
  if (argc!=3) {
    fprintf(stderr, "Invalid number of inputs!\n");
    fprintf(stderr, "Usage: %s file1 file2\n", argv[0]);
    fprintf(stderr, "file1 and file2 have to be of same size.\n");
    exit(1);
  }
  else {
    /* check the file size */
    if ( (stat(argv[1], &stat1)) ) {
      fprintf(stderr, "Cannot open file %s for reading\n", argv[1]);
      exit(2);
    }
    if ( (stat(argv[2], &stat2)) ) {
      fprintf(stderr, "Cannot open file %s for reading\n", argv[2]);
      exit(2);
    }
    if ( stat1.st_size != stat2.st_size ) {
      fprintf(stderr, "file %s and file %s are not the same size.\n",
            argv[1], argv[2]);
      exit(2);
    }
  }
  /* allocating memory for the two files, of course we don't really
need to do
   * that, but I am lazy... */
  if ( (buf1=(char *)malloc(stat1.st_size*sizeof(char))) == NULL ||
       (buf2=(char *)malloc(stat2.st_size*sizeof(char))) == NULL     ) {
    fprintf(stderr, "Cannot allocate memory!\n");
    exit(3);
  }
```

```
 /* open the two files */
  file1 = fopen(argv[1], "rb");
  file2 = fopen(argv[2], "rb");

  /* read the two files */
  fread(buf1, sizeof(char), stat1.st_size, file1);
  fread(buf2, sizeof(char), stat1.st_size, file2);
  fclose(file1);
  fclose(file2);

  /* calculating distortion */
  error = 0.0;
  for (i=0;i<stat1.st_size;i++) {
    x = (unsigned char)(buf1[i]) - (unsigned char)(buf2[i]);
    error += (x * x / stat1.st_size);
  }
  psnr = 10.0 * log10(255*255/error);

  /* output the distortion */
  if ( !strcmp(argv[0], "mse") )
    printf("%g\n", error);
  else if ( !strcmp(argv[0], "psnr") )
    printf("%g\n", psnr);
  else if ( !strcmp(argv[0], "psnrmse") )
    printf("mse: %g PSNR %g\n", error, psnr);

  /* free up memory and exit */
  free(buf1);
  free(buf2);
  return(0);

}
```

Appendix II

// Matlab code for the purpose of performance comparison //

// Matlab code for the purpose of performance comparison //

```
x=0:50:350;
y=12:2:30;
y1=[29.7,23.7,20.7,12];   // gain values for conventional prediction
y2=[29.7,24.2,21.7,13];   // gain values using  criterion  I
y3=[29.7,26,25,16];       //   gain values using  criterion II
x=[0,50,150,350];         // frame number//
plot(x,y1,'-g')
hold on;
plot(x,y2,'-.b');
plot(x,y3,'--r');
hold off;
ylabel('PSNR(db)');
xlabel('frame number');
 title('with no intra updating');
 h=legend('y1=convetional','y2=C-I','y3=C-II');
```

// performance comparison for periodic intra updating //

```
        x=5:5:30;
        y=23:1:30;
y11=[29,28.7,27.8,27.2,26.2];
y12=[29.2,29,28.6,27.4,26.4];
y13=[29.8,29.6,29.3,28.4, 26.8];
x0=[5,10,15,20,30];
plot(x0,y11,'-g');
hold on;
plot(x0,y12,'-.b');
plot(x0,y13,'--r');
hold off;
ylabel('PSNR(db)');
xlabel('packet lossRate in %');
title('with peroidic intra updating');
 h=legend('y11=convetional','y12=C-I','y13=C-II');
 grid on;
```

// RD optimized intra updating//

```
>> x=5:5:30;
y=23:1:30;
y21=[29.2,29,28.2,27.6,26];
y22=[29.8,29.23,28.8,28.2,26.5];
y23=[30,29.6,29.4,28.7,26.8];
x01=[5,10,15,20,30];
plot(x01,y21,'-g');
hold on;
```

```
plot(x01,y22,'-.b');
plot(x01,y23,'--r');
hold off;
ylabel('PSNR(db)');
xlabel('packet lossRate in %');
 h=legend('y21=convetional','y22=C-I','y23=C-II');
title('RD optimized intra updating');
grid on;
 //Matlab program to display the original program & to show the difference between the
                    original and the decompressed//
J=imread('C:\Documents and
          Settings\Administrator\Desktop\h26xCodec\rmd_stdcall\test00000.bmp');
 I=imread('C:\Documents and
          Settings\Administrator\Desktop\h26xCodec\rmd_stdcall\kib00000.bmp');
K=imabsdiff(I,J);       //  computes the difference between image I and  J  //
 imshow(K);             // displays the difference between image I and  J       //
 imshow(J);              // displays the image   J  (decompressed)     //
 imshow(I);               // displays the image   I   (original image )   //
```

Appendix III
CODEC STRUCTURE