EMERGING WIRELESS MULTIMEDIA TECHNOLOGIES AND SERVICES

EMERGING WIRELESS MULTIMEDIA TECHNOLOGIES AND SERVICES

Edited by

A. Salkintzis and N. Passas

A Wiley-Interscience Publication

JOHN WILEY & SONS

New York • Chichester • Weinheim • Brisbane • Singapore • Toronto

Contents

1 Multimedia Coding Techniques for Wireless Networks

Anastasios Delopoulos

Aristotle University of Thessaloniki
Electrical and Computer Engineering Department
Thessaloniki, Greece

1.1 INTRODUCTION

1.1.1 Digital Multimedia and the Need for Compression

All types of information that can be captured by *human perception*, and can be handled by *human made devices* are collectively assigned the term *multimedia content*. This broad definition of multimedia includes text, audio (speech, music), images, video or even other types of signals such as bio-signals, temperature and pressure recordings etc. Limiting the types of human perception mechanisms to vision and hearing senses yields a somehow narrower definition of multimedia content that is closer to the scope of multimedia in our everyday or commercial language. If in addition, we stick to only those information types that can be handled by computer-like devices we come up with the class of *digital multimedia content*. Main representatives of the latter are text, digital audio (including digital speech), digital images and video.

Although some of the concepts presented in this chapter are easily extendable to all types of digital multimedia content, we shall clearly focus to speech and video. The reason of adopting this restrictive approach is that these two modalities (i) constitute the *vast amount of data* transmitted over wireless channels, (ii) they both share the need of been *streamed* through these channels and (iii) much of the *research and standardization* activities have been carried out for their efficient transmission.

Speech is generated by excitation of the vocal track from the air coming out of the lungs in a controlled manner. This excitation produces time varying air pressure in the neighborhood of the mouth in the form of propagating acoustic waves that are possibly captured by human ears. Digital speech is the recording - in the form

of a sequence of samples - of this time varying pressure. A microphone followed by an analog-to-digital converter is performing the recording procedure. Inversely, digital speech can reconverted into acoustic waves by means of an digital-to-analog converter used to excite a speaker. More detail regarding the above procedures can be found in Section 1.3.1.

Many design parameters influence the quality of digital speech, that is how accurately the reverse procedure can reproduce the original (analog) speech acoustic waves. The most important between them is the sampling frequency (how often pressure is being measured) and the accuracy in the representation of each single sample, namely the number of quantization levels used. The latter is closely related to the number of bits used for the discrete representation of the samples.

Typical choices for low quality digital speech include sampling at 8000 samples per second with 8 bits per sample. This sums up to 64000 bits per second, a bitstream rate that is well above traditional voice communication channels. The situation becomes even harder considering that this rate does not include neither error correction bits nor signaling overhead. The need of compression is apparent.

Unlike speech, video has not a structured generation mechanism (the analogous of vocal track). On the contrary, its capturing mechanism is well defined. We may think of video as the time varying recording of light (both luminance and color) on the cells of eye retina. In fact this light corresponds to the idol (image) of viewed scenes as produced on the retina by means of the eye-lenses. The human visual perception mechanism is briefly explained in Section 1.6.1. Digital video is an approximation of the eye perceived information in the form of a three dimensional sample sequence. Two of the dimensions correspond to the location of each sample w.r.t the idol coordinate system while the third corresponds to the time instance each sample has been measured. In its simplest form, digital video can be considered as a sequence of still digital images (frames) which in turn are fixed-sized two dimensional arrays of samples (light recordings).

Temporal sampling frequency (frame rate), the dimension of frames and the number of quantization levels for each sample determine the quality of digital video and the associated bitstream rate. Typical medium quality choices include 25 frames per second, 576×720 frame dimensions and 24 bits per sample witch translates to 237 MegaBytes per second, a bitrate that is far beyond the capacities of available communication channels. Compressing digital video is thus necessary if it is to be transmitted.

In view of the previous considerations it is not surprising that much of the effort of the signal processing community has been devoted to devising efficient compression-decompression algorithms for digital speech and video. A range of standardization bodies is also involved since the produced algorithms are to be used by diverse manufacturers and perhaps in cross platform environments. This fruitful combined effort has already produced a collection of *source coders-decoders* (an alternative name for compression-decompression algorithms for multimedia content). Most of these *codecs* are already components of today's multimedia communication products. As

expected, though, the research is ongoing mainly busted by the improved processing capabilities of the emerging hardware.

Codec design is guided by a number of, sometimes antagonistic, requirements / specifications:

Targeted quality: This is determined by the targeted application environment; quality of video in entertainment applications such as digital television or theaters is certainly more demanding than that in video-conference applications.

Targeted bitrate: This is mainly determined by the medium used to transmit (or store) the compressed multimedia representations. Transmission of speech over person to person wireless communication channels is usually much more bandwidth parsimonious than its counterpart in wired or broadcast environments.

Targeted Complexity and Memory Requirements: This is mainly determined by the type of the device (hardware) that hosts the codec and is also related to the power consumption constraints imposed by these devices.

1.1.2 Standardization activities

Multimedia coding techniques are the subject of standardization activities within multi/inter-national organizations (ITU, ISO, ETSI, 3GPP2) and national authorities (e.g., U.S. Defense Office) and associations (North American TIA, Japanese TTC, etc.)

The *International Telecommunication Union (ITU)* contributes to the standardization of multimedia information via its Study Group 15 and Study Group 16 including *Video Coding Experts Group (VCEG)*. A number of standardized speech/audio codecs are included in series G.7xx (Transmission systems and media, digital systems and networks) of ITU-T Recommendations. Video coding standards belong to series H.26x (Audiovisual and multimedia systems).

The *International Organization for Standardization (ISO)* and particularly its *Motion Pictures Experts Group (MPEG)* has produced a series of widely accepted video and audio encoding standards like the well known MPEG-1, MPEG-2 and MPEG-4.

The *European Telecommunications Standards Institute (ETSI)* is an independent, non-profit organization, producing telecommunications standards in the broad sense. A series of speech codecs for GSM communications have been standardized by this organization.

The *Third Generation Partnership Project 2 (3GPP2)* is a collaborative third generation (3G) telecommunications specifications-setting project comprising North American and Asian interests. Its goal is to develop global specifications for ANSI/TIA/EIA-41 Cellular Radiotelecommunication Intersystem Operations network evolution to 3G and global specifications for the Radio Transmission Technologies (RTTs). The following telecommunication associations participate in 3GPP2:

1. *ARIB*: Association of Radio Industries and Businesses (Japan)
2. *CCSA*: China Communications Standards Association (China)
3. *TIA*: Telecommunications Industry Association (North America)
4. *TTA*: Telecommunications Technology Association (Korea)
5. *TTC*: The Telecommunication Technology Committee (Japan)

1.1.3 Structure of the chapter

The remaining of the chapter contains three main parts.

At first, Section 1.2 offers a unified description of multimedia compression techniques. The fundamental ideas of Entropy Coding, Redundancy Reduction and Controlled Distortion as tools for reducing the size of multimedia representations are introduced in this section.

The second part is devoted to speech coding. Sections from 1.3 to 1.5 belong to this part. The nature of speech signals is explored first in order to validate the synthetic models representing next. Specialization of the general compression ideas in the field of speech coding is presented leading to the presentation of most important speech coding algorithms including the popular CELP. The last section of this part refers to speech codecs standardized by the ITU, the ETSI and the 3GPP2 linking them to the aforementioned speech coding algorithms.

The third part consists of Sections 1.6.3 and 1.7 and covers digital video compression aspects following a similar structure. The nature of video signals is explored first and the adaptation of general compression techniques to the case of digital video is considered next. Most important digital video compression standards are presented in the last section.

A bibliography section completes present chapter.

1.2 BASICS OF COMPRESSION

Compression of speech, audio and video steps on the nature of these signals, modelling of their generation mechanism (e.g., vocal track for speech) and/or exploitation of human perception limits (e.g., audible spectrum, tone masking, spatiotemporal visual filters).

1.2.1 Entropy, Entropy Reduction and Entropy Coding

In their digital form, multimedia modalities can be considered as streams of symbols which are produced by the quantizer. All of them are picked from a finite set $S = \{ s_0, s_1, \ldots, s_{N-1} \}$ where N coincides with the number of quantization levels used in the discrete representation. Clearly more accurate representations

require more quantization levels (finer quantizer) and thus higher values of N. Each symbol s_i may appear in the stream with a certain probability $p_i \in [0, 1]$. In *Information Theory* language the mechanism that generates the particular stream and the corresponding symbols is called *symbol source* and is totally characterized by the set of probabilities p_i, $i = 0, \ldots, N-1$. It turns out [1] that the non-negative quantity,

$$H_s = -\sum_{i=0}^{N-1} p_i \, log_2(p_i) \tag{1.1}$$

called *entropy* determines the lower bound of the number of bits that are necessary in order to represent the symbols produced by this particular source.

More specifically, let $W = \{ w_0, w_1, \ldots, w_{N-1} \}$ be the N different binary (containing 0's and 1's) words used to represent the symbols of S and $|w_i|$ the number of bits of each word (length); the average number of bits used to represent each symbol is,

$$L_w = \sum_{i=0}^{N-1} p_i \, |w_i| \, . \tag{1.2}$$

It should be clear that L_w multiplied by the rate of symbol production (symbols per second), a parameter that is not essentially controlled by the codec, yields the bitstream rate.

It can be proved that always (even for the cleverest choice of w_i's)

$$L_w \geqslant H_s \, . \tag{1.3}$$

In view of equations (1.1), (1.2) and (1.3) the design of a good codec reduces to:

1. Transforming the original sample sequences into an alternative sequence, whose samples can be quantized by symbols in S', with entropy $H_{s'}$ lower than the original H_s.
2. Cleverly adopting a binary word representation, W, that has average word length, L_w, as close to the lower bound $H_{s'}$ as possible.

If the aforementioned transformation is perfectly reversible, the codec is characterized as *lossless*. In the opposite case, i.e., when the inverse transformation results to an inperfect approximation of the original signal, the codec is *lossy*. Lossy codecs are pretty popular in multimedia coding scenarios since they may result in significant reduction of entropy and thus allow for extremely low L_w's and correspondingly for low bitstream rates.

On the other hand, the aforementioned clever selection of binary words for the representation of symbols is a procedure called *entropy coding* which does not introduce any information loss. Celebrated representatives of entropy coding techniques are among others, the *Huffman* entropy coder [2] and the *algebraic* coder (see e.g., [3]).

Present chapter does not include any further detail regarding entropy coding. On the contrary, our attention is focusing on lossy procedures for the alteration of the

original speech and video signals into lower entropy representations since there is the emphasis of (and the differentiation among) source codecs used in wireless multimedia communications.

Methods for transforming the original speech or video signals into sample sequences of lower entropy fall into two complementary categories, namely, *controlled distortion* and *redundancy reduction*.

1.2.1.1 Controlled Distortion Some portions of multimedia signals' content are less important than others in terms of hearing or visual perception. Suppressing these parts may reduce contents entropy without affecting signals *perceptual quality*. In addition, although some types of distortion are audible or visible, they are considered as unimportant provided they do not alter signals' semantics; e.g., they do not cause phoneme /a/ be confused with /e/, etc.

1.2.1.2 Redundancy Reduction Redundancy in source coding is synonymous to temporal or spatial correlation.

Consider for example a video sequence produced by a stationary camera capturing a stationary scene. The first captured frame is (almost) identical to all subsequent ones; a redundancy reduction procedure in this simple case would be to encode the first frame and simply inform the decoder that the depicted scene is not changing for a certain time interval!

In the case of speech coding consider a speech segment that is identical to a sinusoid. Provided that the frequency and phase of the sinusoid is identified it suffices for coding purposes to encode a single sample of the signal and the frequency and phase values. All remaining samples can be recovered using this information.

Although both aforementioned examples are too simplistic they validate our initial argument that links redundancy to correlation. Using pure mathematical arguments the following principle can be established:

Let $y(n) = F[x(n)]$ be a reversible transform of signal $x(n)$ and $s_x(n)$, $s_y(n)$ the quantized versions (symbols) of the original and the transformed signals. If the correlation $E\{y(n)y(n+m)\} < E\{x(n)x(n+m)\}$ then the entropy of the 'source' $s_y(n)$ is less than the entropy of $s_x(n)$.

Application of this principle is extensively used in speech and video coding. Correlation reduction methods include (i) whitening by prediction (see e.g. DPCM and ADPCM methods) used to reduce temporal correlation, (ii) Motion estimation and compensation used to reduce spatiotemporal correlation in video coding (see e.g. MPEG coding), (iii) Transform coding used to reduce spatial correlation in video encoding (see e.g. H.263 and MPEG algorithms).

1.2.2 A general compression scheme

Following the previous discussion, compression methods contain the stages of the block diagram of Figure 1.1. Inclusion of the grayed blocks characterizes lossy compression schemes while in lossless compressors these blocks are not present.

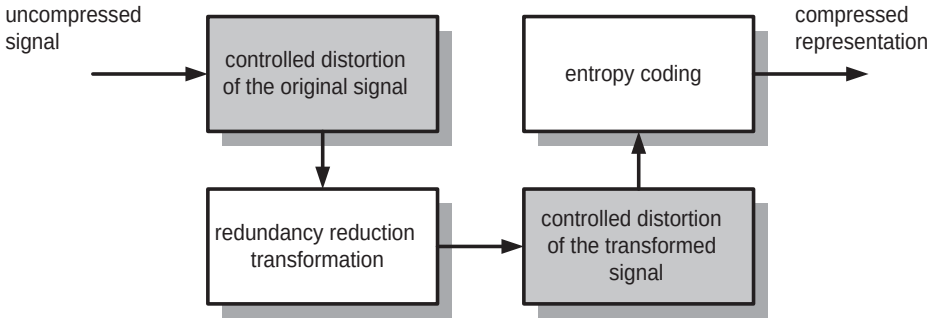

Fig. 1.1 Basic steps in multimedia source compression

1.3 UNDERSTANDING SPEECH CHARACTERISTICS

1.3.1 Speech Generation and Perception

Speech is produced by a cooperation of lungs, glottis (with vocal cords) and articulation tract (mouth and nose cavity). Speech sounds belong to two main categories: voiced and unvoiced.

Voiced sounds (like $/a/$ in *bat*, $/z/$ in *zoo*, $/l/$ in *let* and $/n/$ in *net*) are produced when the air coming out of the lungs through the epiglottis, causes a vibration of the vocal cords which in turn interrupt the outgoing air stream and produce an almost periodic pressure wave. The pressure impulses are commonly called *pitch impulses* and the fundamental frequency of the pressure signal is the pitch frequency or simply the *pitch*. Pitch impulses excite the air in the mouth and/or nose cavity. When these two cavities resonate they radiate the speech sound with resonance frequency content depending upon their particular shape. The principal resonance frequencies of voiced speech sounds are called *formants*, they may change by reshaping mouth and noise cavity and determine the type of the produced *phoneme*.

On the other hand unvoiced speech is characterized by the absence of periodic excitation (pitch impulses). An unvoiced sound wave has either the characteristics of a *noise signal* (like $/s/$ in *sun*)produced by the *turbulent flow* of the air through the complex of teeth and lips or the form of a response of the vocal tract to a sudden (impulsive) release of air (like $/p/$ in *put*)

1.3.2 Digital Speech

Human ear acts as a bandpass filtering mechanism. Only frequencies in the range of 20 to $20,000$ Hz excite the earing system of average listeners. Although, human voice may reach or even exceed the upper limits of this range, it has been experimentally recognized that most of the semantics within speech are carried in a portion of frequency range extending from 200 to $3,200$ Hz while almost the entire speech content is included within 50 to 7.000 Hz.

Following these observations Digital Speech is produced by A/D converters sampling at either $8,000$ samples/sec (for telephone quality speech) or $16,000$ samples/sec (for high quality speech). Both sampling frequencies satisfy the requirements of Nyquist theorem i.e., they are more than the double of the acquired signals maximum frequency. Nevertheless, higher sampling frequencies are adequate for capturing high quality music (including voice).

As far as quantization is concerned, speech samples are quantized with either 8 or 16 bits/sample.

Three types of quantizers are usually employed:

Linear Quantizer: Speech values are normalized in the range of $[-1,\ 1]$ and uniform quantization to 2^8 or 2^{16} levels is performed next.

A-law Quantizer: Original sample values, x, are first transformed according to a logarithmic mapping

$$y = \begin{cases} \frac{A|x|}{1+\log A} sign(x) & \text{for } 0 \leqslant |x| \leqslant \frac{V}{A} \\ \frac{V(1+\log(A|x|/V))}{1+\log A} sign(x) & \text{for } \frac{V}{A} < |x| \leqslant V \end{cases} \tag{1.4}$$

and the resulting y values are quantized by a uniform 8-bit quantizer. In the above formula, V is the peak value of $|x|$ and A determines the exact decision levels of the A-law quantizer (typical value for $A = 87.6$).

A-law quantizers result in more accurate representation of low valued samples.

μ-law Quantizer: Similarly to A-law, original sample values, x, are mapped to

$$y = \frac{V \log(1 + \mu|x|/V)}{\log(1 + \mu)} sign(x), \tag{1.5}$$

and 8-bit uniform quantization is performed next. V is again the peak value of $|x|$ and parameter μ determines the exact values of the decision levels.

Both A-law and μ-law quantizers are prescribed in G.711 ITU standard ([4]) and further details can be found in [5].

1.3.3 Speech Modelling and Linear Prediction

Voiced and unvoiced speech are modelled as the output of a linear filter excited by an impulse sequence or a white noise signal respectively. The effect of the vocal tract is approximated by a gain followed by a linear filter, $H(z)$. The exact type of the involved filter is significantly different for the two types of speech. Within each type the filter parameters determine the phoneme to be modelled.

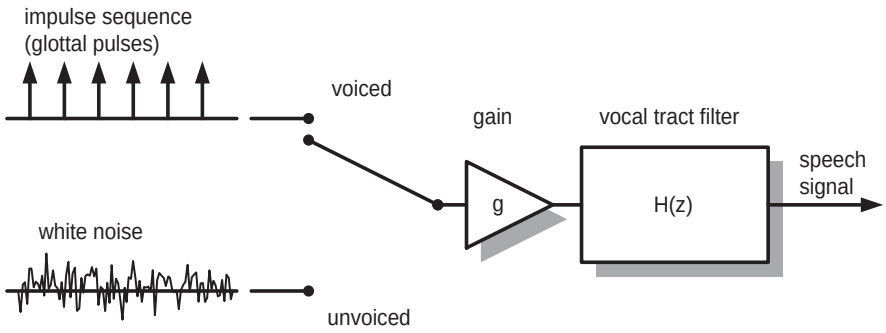

Fig. 1.2 The fundamental model for Voiced and Unvoiced speech production

In order to imitate the voiced speech generation mechanism, as described in Section 1.3.1, the linear filter $H(z)$ of Figure 1.2 is the cascade of two simpler filters ($H(z) = H_V(z) \equiv H_m(z)H_e(z)$). The first, $H_e(z)$, is a low-pass filter approximating the effect of epiglottis that transforms the input impulse sequence into a pulse sequence. The second, $H_m(z)$, approximates the resonance effect of mouth and noise. The latter is responsible for raising formant frequencies by reshaping the spectrum of incoming signal.

In the case of unvoiced speech, the random noise sequence is passed through a single filter, $H(z) = H_U(z)$, that approximates the effects of teeth and lips.

Many compressors employed in wireless speech communication make extensive use of the presented speech modelling schemes. Based on the spectral characteristics of a particular speech segment these coders try to estimate (a) the characteristics of input excitation sequence (i.e., the pitch for voiced and noise characteristics for unvoiced speech), (b) the gain g and (c) the parameters of $H_V(z)$ or $H_U(z)$ for voiced and unvoiced speech segments respectively. Having available the estimates of the speech model that best fits the speech segment in hand the coders act in either of two ways: (i) Use the model as a linear predictor of the true speech (see e.g., ADPCM methods in Section 1.4.1.3) or (ii) Encode and transmit the model's parameters rather than speech samples themselves (see Sections 1.4.2 and).

A variety of signal processing algorithms have been adopted by different codecs for appropriately fitting $H_V(z)$ and/or $H_U(z)$ to the speech segment under consideration.

All of them fall into the family of Linear Prediction (LP) methods. They attempt to estimate the parameters of an Autoregressive (AR) Model,

$$H(z) = \frac{1}{1 - A(z)} = \frac{1}{1 - \sum_{i=1}^{q} a_i z^{-i}} \tag{1.6}$$

that when feeded with a particular input, $x(n)$, produces an output, $y(n)$, as close as possible to a given signal (desired output), $s(n)$. In speech coding applications the input is either an impulse train with period equal to the pitch (for voiced speech) or a white noise sequence (for unvoiced speech). The desired output, $s(n)$, is the speech segment at hand.

In time domain, filtering by $H(z)$ is equivalent to the recursion

$$y(n) = \sum_{i=1}^{q} a_i y(n-i) \; + \; x(n) \tag{1.7}$$

Optimal selection of $\{a_i\}$, $(i = 1 \ldots q)$ AR coefficients is based on the minimization of the Mean Squared Error (MSE) Criterion,

$$J = E\{(s(n) - y(n))^2\} \tag{1.8}$$

which is approximated by its deterministic counterpart

$$J_N = \frac{1}{N} \sum_{n=0}^{N-1} \{(s(n) - y(n))^2\} \tag{1.9}$$

where N is the number of the available (speech) samples $s(n)$, $(n = 0, \ldots, N-1)$. An exhaustive review of algorithms for estimating a_i can be found in [6].

In the case of unvoiced speech, the assumption of white noise input $x(n)$ is enough for the estimation of the AR parameters. On the other hand, for the case of voiced speech the pitch period, p, should be estimated prior to a_i's estimation. This requires an extra effort and is usually based on the analysis of the autocorrelation sequence $r_{ss}(m) \equiv E\{s(n)s(n+m)\}$ of $s(n)$. Periodicity of $s(n)$ causes periodicity of $r_s(m)$ as well with the same period p equal to the pitch period. Hence determination of p reduces to finding the first dominant peak of $r_s(m)$ that occurs at lag $m = p > 0$. As an alternative, a rough estimate, $\overline{p}$, of p can be adopted and the exact value of p is chosen in a way that minimizes J_N among different candidate values of p in the neighborhood of $\overline{p}$.

1.3.4 General Aspects of Speech Compression

1.3.4.1 Controlled Distortion of Speech

High frequencies of audio signals: Average human ear acts as a pass-band filter cutting off frequencies out of the range of $20 - 20.000$ Hz. Thus, almost perfect representation of any audio signal (including speech, music, etc.) is achieved even if we filter out the corresponding frequency components prior to analog-to-digital conversion. This reduces the effective bandwidth of audio signals and allows for sampling rates as low as $2 \times 20.000 = 40.000$ samples per second in conformance to the Nyquist Theorem [7].

High frequencies of speech signals: Speech signals content is mostly concentrated in an even shorter bandwidth range, namely $50 - 7.000$ Hz which indicates that pre-filtering the remaining frequency components allows for sampling rates as low as 14.000 samples per second. In fact, speech content outside the range of $200 - 3.200$ Hz is hardly affecting speech semantics which means that sampling at 6.400 samples per second is sufficient at least for (low quality) human communication. Based on this observation most speech coding schemes use sampling at the rate of 8.000 samples per second.

Accuracy of sample representation: It has been experimentally justified that 2^{16} quantization levels are more than satisfactory for representing speech samples. Thus, finer sample variation is omitted from speech source encoding schemes and 16 bits per sample are usually used at the quantization stage.

Tonal masking: Psychoacoustic experiments have justified that existence of strong tonal components, corresponding to sharp and high peaks of the audio spectrum, makes human ear insensitive to frequencies with lower power in the neighborhood of the tone. The tone is characterized as *masker* while the neighboring frequencies are *masked*. After this observation efficient audio codecs divide the original signal into sub-bands with narrow bandwidth and use fewer bits (leading to increased but still not audible quantization error) for those bands that are masked by strong maskers. MPEG audio layer I-II (see e.g., [8]) and layer III (see e.g., [9]) codecs make extensive use of such type of controlled distortion (ref. Section 1.4.1.4). Also, Analysis by Synthesis speech codecs take into account the masking effect of high energy frequency components (formants) for selecting perceptually optimal parametric models of the encoded speech segments (ref. Section 1.4.3).

1.3.4.2 Redundancy Reduction of Speech Redundancy of speech signals is due to the almost periodic structure of their voiced parts. Speech codecs attempt to model this periodicity using Autoregressive filters whose coefficients are calculated via Linear Prediction methods as explained in Section 1.3.3. Having available these parametric models redundancy reduction is achieved by two alternative approaches:

1. Decorrelating speech via prediction as described in Section 1.4.1.3 (ADPCM coding).

2. Encoding model parameters instead of the samples of the corresponding speech segment. This approach is the core idea of Analysis-Synthesis and Analysis-By-Synthesis codecs presented in Sections 1.4.2 and 1.4.3.

1.4 THREE TYPES OF SPEECH COMPRESSORS

1.4.1 Waveform compression

Waveform compression refers to those codecs that attempt to transform digitized speech sequences (waveforms) into representations that require less bits (lower entropy). Main objective is to be able to reconstruct the original waveform with as low error as possible. The internal structure (pitch, formant) of speech is ignored. Evaluation of the error may though take into account the subjective properties of human ear. Most of algorithms of this type are equally applicable to general audio signals (e.g., music).

1.4.1.1 Pulse Code Modulation (PCM) PCM is the simplest speech (actually general audio) codec that basically coincides with the couple of a sampler and a quantizer. Bitrate is determined by controlling only the sampling rate and the number of quantization levels.

The input analog speech signal $x_0(t)$ is first passed through a low pass filter of bandwidth B to prevent aliasing. The output $x(t)$ of this antialising filter is next sampled:

$$x(n) = x(nT)$$

with sampling frequency $f_s = 1/T < B/2$ and quantized by some scalar quantizer Q[]:

$$s(n) = Q[x(nT)]$$

Most PCM speech coders use a sampling frequency f_s of 4000 or 8000 samples per second. On the other hand, quantization uses 16 bits or less for each symbol.

1.4.1.2 Differential Pulse Code Modulation (DPCM) Differential Pulse Code Modulation is the oldest compression scheme that attempts to reduce entropy by removing temporal correlation (ref. Section 1.2.1.2). Similarly to PCM, a sequence of samples $x(n)$ are initially produced after filtering and sampling. A differential signal is next produced as,

$$d(n) = x(n) - x(n-1). \tag{1.10}$$

Clearly the original sequence can be reproduced from the initial sample $x(0)$ and the sequence $d(n)$ by recursively using

$$x(n) = x(n-1) + d(n) \quad \text{for} \quad n = 1, \ldots \tag{1.11}$$

The idea behind coding sequence $d(n)$ instead of $x(n)$ is that usually $d(n)$ is less correlated and thus according to the observation of Section (1.2.1.2) it assumes lower entropy. Indeed, assuming without loss of generality that $E\{x(n)\} = 0$, autocorrelation $r_d(m)$ of $d(n)$ can be calculated as follows:

$$\begin{aligned} r_d(m) &= E\{d(n)d(n+m)\} \\ &= E\{(x(n)-x(n-1))(x(n+m)-x(n+m-1))\} \\ &= E\{x(n)x(n+m)\}+E\{x(n-1)x(n+m-1)\} \\ &-E\{x(n)x(n+m-1)\}-E\{x(n-1)x(n+m)\} \\ &= 2r_x(m)-r_x(m-1)-r_x(m+1) \\ &\approx 0, \end{aligned} \tag{1.12}$$

where in the last row of (1.12) we used the assumption that the autocorrelation coefficient $r_x(m)$ is very close to both $r_x(m-1)$ and $r_x(m+1)$. In view of (1.12) we may expect that under certain conditions (not always though) the correlation between successive samples of $d(n)$ is low even in the case that the original sequence $x(n)$ is highly correlated. We thus expect that $d(n)$ has lower entropy than $x(n)$.

In practice the whole procedure is slightly more complicated because $d(n)$ should be quantized as well. This means that the decoder cannot use (1.11) as is since that would result in accumulation of quantization error. For this reason the couple of expressions (1.10), (1.11) is replaced by:

$$d(n) = x(n) - \widehat{x}(n-1), \tag{1.13}$$

where

$$\begin{aligned} \widehat{x}(n) &= \widehat{d}(n) + \widehat{x}(n-1) \\ \widehat{d}(n) &= \overline{Q}[d(n)]. \end{aligned} \tag{1.14}$$

DPCM as already described is essentially an one-step ahead prediction procedure, namely $x(n-1)$ is used as a prediction of $x(n)$ and the prediction error is next coded. This procedure can be generalized (and enhanced) if prediction takes into account more past samples weighted appropriately in order to capture signal's statistics. In this case, equations 1.10 and 1.11 are replaced by their generalized counterparts:

$$\begin{aligned} d(n) &= x(n) - \mathbf{a}^T\mathbf{x}(n-1) \\ x(n) &= d(n) + \mathbf{a}^T\mathbf{x}(n-1) \end{aligned} \tag{1.15}$$

where sample vector $\mathbf{x}(n-1) \stackrel{\triangle}{=} [x(n-1)\ x(n-2)\ \cdots\ x(n-p)]^T$ contains p past samples and $\mathbf{a} = [a_1\ a_2\ \cdots\ a_p]^T$ is a vector containing appropriate weights known also as *prediction coefficients*.

Again in practice (1.15) should be modified similarly to (1.14) in order to avoid accumulation of quantization errors.

1.4.1.3 Adaptive Differential Pulse Code Modulation (ADPCM) In the simplest case, prediction coefficients $\mathbf{a}$ used in (1.15) are constant quantities characterizing the particular implementation of the (p-step) DPCM codec. Better decorrelation of $d(n)$ can be achieved, though, if we *adapt* these prediction coefficients to the particular correlation properties of $x(n)$. A variety of batch and recursive methods can be employed for this task resulting to the so called Adaptive Differential Pulse Code Modulation (ADPCM).

1.4.1.4 Perceptual audio coders (MPEG layer III (MP3) etc) Both DPCM and ADPCM exploit *redundancy reduction* to lower entropy and consequently achieve better compression than PCM. Apart from analog filtering (for antialising purposes) and quantization, they do not distort the original signal $x(n)$. On the opposite side, the family of codecs of this section applies serious controlled distortion to the original sample sequence in order to achieve far lower entropy and consequently much better compression ratios.

Perceptual audio coders, with most celebrated representative the MPEG-1 layer III audio codec (MP3) (standardized in ISO/IEC 11172-3, [10]), split the original signal into subband signals and for each subband use quantizers of different quality depending on the perceptual importance of each subband.

Perceptual coding relies on four fundamental observations validated by extensive psychoacoustic experiments:

1. Human hearing system cannot capture *single tonal* audio signals (i.e., signals of narrow frequency content) unless their power exceeds a certain threshold. The same holds also for the distortion of audio signals. The aforementioned audible threshold depends on the particular frequency but is relatively constant among human listeners. Since this threshold refers to single tones at the absence of other audio content it is called audible threshold in quite (ATQ). A plot of ATQ versus frequency is presented in Figure 1.3.

2. An audio tone of high power, called masker, causes an increase of the audible threshold for frequencies close to its own frequency. This increase is higher for frequencies close to the masker, and decays according to a spreading function. A plot of Audible Threshold in the presence of a masker is presented in Figure 1.4.

3. Human ear perceives frequency content in almost logarithmic scale. Bark scale rather than linear frequency (Hz) scale is more representative of the ear's ability to distinguish between two neighboring frequencies. Bark frequency z is usually calculated from its linear counterpart f as:

$$z(f) = 13 \arctan(0.00076 f) + 3.5 \arctan\left(\left(\frac{f}{7500}\right)^2\right) (Bark)$$

   Figure 1.5 illustrates the plot of z versus f. As a consequence the aforementioned masking spreading function has an almost constant shape when it is

expressed in terms of barks frequency. In terms of linear frequencies (Hz) this leads to wider spreading for maskers with (linear) frequencies residing close to the upper end of the audible spectrum.

4. By dividing the audible frequency range into bands of one bark width we get the so called critical bands. Concentration of high power noise (non tonal audio components) within one critical band causes an increase of the audible threshold of the neighboring frequencies. Hence, such concentrations of noise resemble the effect of tone maskers and are called Noise Maskers. Their masking effect spreads around their central frequency in a manner similar to their tone counterpart.

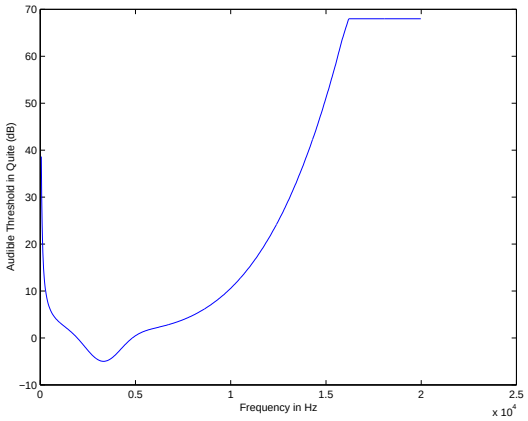

Fig. 1.3 Audible Threshold in quite vs. frequency in Hz

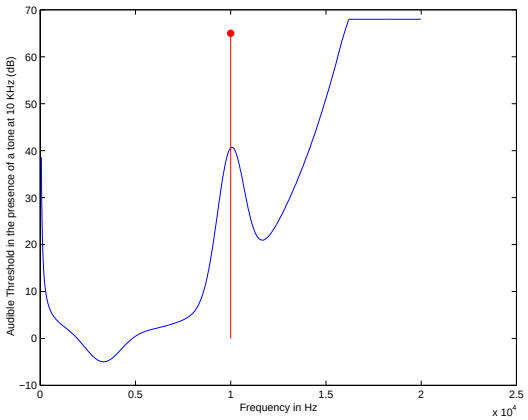

Fig. 1.4 Audible Threshold in the presence of a 10 KHz tone vs. frequency in Hz

Based on these observations, Perceptual Audio Coders (i) sample and finely quantize the original analog audio signal, (ii) segment it into segments of approximately 1 sec duration, (iii) transform each audio segment into an equivalent frequency representation employing a set of complementary frequency selective subband filtters (subband analysis filterbank) followed by a modified version of Discrete Cosine Transform (M-DCT) block , (iv) estimate the overall audible threshold, (v) quantize the frequency coefficients trying to keep quantization errors just under the corresponding audible threshold. The reverse procedure is performed at decoder's side.

A thorough presentation of the details of Perceptual Audio Coders can be found in [11] or [9] while the exact encoding procedure is defined in ISO standards [MPEG audio layers I, II, III].

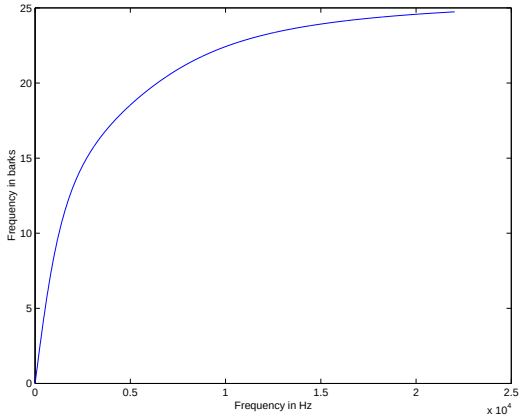

Fig. 1.5 Barks number vs. frequency in Hz

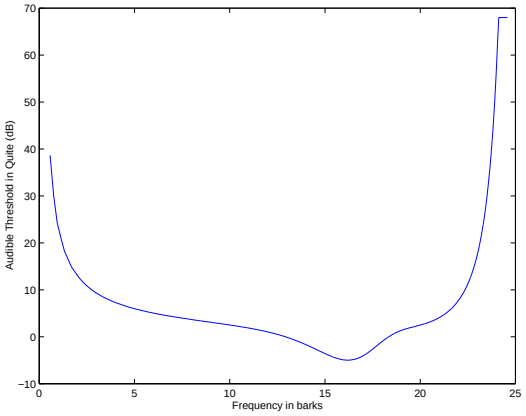

Fig. 1.6 Audible Threshold in quite vs. frequency in Barks

1.4.2 Open-Loop Vocoders / Analysis - Synthesis Coding

As explained in the previous section Waveform Codecs share the concept of attempting to approximate the original audio waveform by a copy that is (at least perceptually) close to the original. The achieved compression is a result of the fact that by design the copy has less entropy than the original.

Open-Loop Vocoders (see e.g., [12]) of this section and their Closed-Loop descendants, presented in the next section, share a pretty different philosophy initially introduced by H. Dudley in 1939 [13] for encoding analog speech signals: Instead of approximating speech waveforms, they try to dig out models (in fact digital filters) that describe speech generation mechanism. The parameters of these models are next coded and transmitted. The corresponding encoders are then able to re-synthesize speech by appropriately exciting the prescribed filters.

In particular, Open-Loop Vocoders rely upon voiced/unvoiced speech models and use representation of short time speech segments by the corresponding model parameters. Only (quantized versions of) these parameters are encoded and transmitted. Decoders approximate the original speech by forming digital filters on the basis of the received parameter values and exciting them by pseudo-random sequences. This type of compression is highly efficient in terms of compression ratios, have low encoding and decoding complexity at the cost of low reconstruction quality.

1.4.3 Closed-Loop Coders / Analysis by Synthesis Coding

This type of speech coders is the preferred choice for most wireless systems. They exploit the same ideas with the Open-Loop Vocoders but improve their reconstruction quality by encoding not only speech model parameters but also information regarding the appropriate excitation sequence that should be used by the decoder. A computationally demanding procedure is employed at encoder's side in order to select the appropriate excitation sequence. During this procedure encoder imitates decoder's synthesis functionality in order to select the optimal excitation sequence from a pool of predefined such sequences (known both to the encoder and the decoder). Optimal selection is based on the minimization of audible (perceptually important) reconstruction error.

Figure 1.7 illustrates the basic blocks of an Analysis-by-Synthesis speech encoder. The speech signal $s(n)$ is approximated by a synthetically generated signal $s_e(n)$. The latter is produced by exciting the cascade of two autoregressive (AR) filters with an appropriately selected excitation sequence. Depending on the type of the encoder this sequence is either selected from a predefined pool of sequences or dynamically generated during the encoding process. The coefficients of the two AR filters are chosen so that they imitate the natural speech generation mechanism. The first, is a *long term predictor* of the form

$$H_L(z) = \frac{1}{1 - A_L(z)} = \frac{1}{1 - az^{-p}} \tag{1.16}$$

in the frequency domain, or

$$y(n) = a\, y(n-p) \,+\, x(n)\,, \tag{1.17}$$

in the time domain, that approximates the pitch pulse generation. The delay p in (1.16) corresponds to the pitch period. The second filter, a *short term predictor* of the form,

$$H_S(z) = \frac{1}{1 - A_S(z)} = \frac{1}{1 - \sum_{i=1}^{K} a_i z^{-i}} \tag{1.18}$$

shapes the spectrum of the synthetic speech according to the formant structure of $s(n)$. Typical values of filter order K are in the range 10 to 16.

Encoding of a speech segment reduces to computing / selecting (a) the AR coefficients of $A_L(z)$ and $A_S(z)$, (b) the gain g and (c) the exact excitation sequence. Selection of the aforementioned optimal parameters is based on minimizing the error sequence $e(n) = s(n) - s_e(n)$. In fact, the Mean Squared Error (MSE) of a weighted version $e_w(n)$ is minimized where $e_w(n)$ is the output of a filter $W(z)$ driven by $e(n)$. This filter that is also dynamically constructed (as a function of $A_S(z)$) imitates human hearing mechanism by suppressing those spectral components of $e(n)$ that are close to high energy formants (ref. Section 1.4.1.4 for perceptual masking behavior of the ear).

Analysis-by-Synthesis coders are categorized by the exact mechanism they adopt for generating the excitation sequence. Three major families will be presented in the sequel: (a) the Multi-Pulse Excitation model (MPE) , (b) the Regular Pulse Excitation model (RPE) and (c) the Vector or Code Excited Linear Prediction model (CELP) and its variants (ACELP, VSELP).

1.4.3.1 Multi-Pulse Excitation coding - MPE This method was originally introduced by Atal and Remde in [14]. In its original form MPE was using only short term prediction.The excitation sequence is a train of K unequally spaced impulses

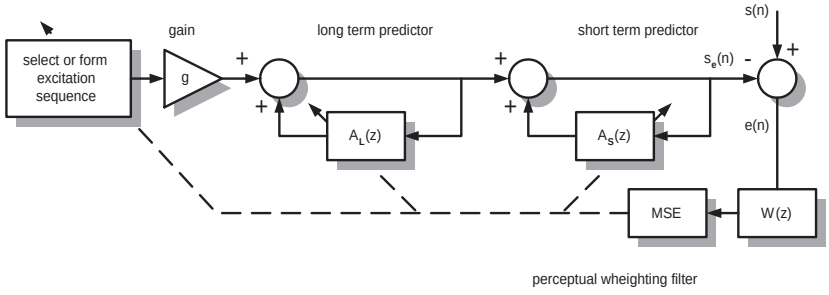

Fig. 1.7 Basic blocks of an Analysis-by-Synthesis speech encoder

of the form

$$x(n) = x_0\delta(n-k_0) + x_1\delta(n-k_1) + \ldots + x_{K-1}\delta(n-k_{K-1})\,, \qquad (1.19)$$

where $\{k_0,\, k_1,\, \ldots,\, k_{K-1}\}$ are the locations of the impulses within the sequence and x_i ($i = 0, \ldots, K-1$) the corresponding amplitudes. Typically K is 5 or 6 for a sequence of $N = 40$ samples (5 ms at 8000 samples/sec). The impulse locations k_i and amplitudes x_i are estimated according to the minimization of the perceptually weighted error, quantized and transmitted to the decoder along with the quantized versions of the short term prediction AR coefficients. Based on this data the decoder is able to reproduce the excitation sequence and pass it through a replica of the short prediction filter in order to synthetically generate an approximation of the encoded speech segment.

In more detail, for each particular speech segment, the encoder performs the following tasks:

Linear Prediction: The coefficients of $A_S(z)$ of the model in (1.18) are first computed employing Linear Prediction (ref. end of Section 1.3.3).

Computation of the Weighting Filter: The employed weighting filter is of the form

$$W(z) = \frac{1 - A_S(z)}{1 - A_S(z/\gamma)} = \frac{1 - \sum_{i=1}^{10} a_i z^{-i}}{1 - \sum_{i=1}^{10} \gamma^i a_i z^{-i}}, \qquad (1.20)$$

where γ is a design parameter (usually $\gamma \approx 0.8$). The transfer function of $W(z)$ of this form has minima in the frequency locations of the formants i.e., the locations where $|H(z)|_{z=e^{i\omega}}$ attains its local maxima. It thus suppresses error frequency components in the neighborhood of strong speech formants; this behavior is compatible with the human hearing perception.

Iterative estimation of the optimal multipulse excitation: An all-zero excitation sequence is assumed first and in each iteration a single impulse is added to the sequence so that the weighted MSE is minimized. Assume that $L < K$ impulses have been added so far with locations $k_0, \ldots, L-1$. The location and amplitude of the $L+1$ impulse are computed based on the following strategy: If $s^L(n)$ is the output of the short time predictor excited by the already computed L-pulse sequence and k_L, x_L the unknown location and amplitude of the impulse to be added, then

$$s^{L+1}(n) = s^L(n) + h(n) \star x_L\delta(n-k_L)$$

and the resulting weighted error is

$$e_W^{L+1}(n) = e_W^L(n) - h_\gamma(n) \star x_L\delta(n-k_L) = e_W^L(n) - x_L h_\gamma(n-k_L) \qquad (1.21)$$

where $e_W^L(n)$ is the weighted residual obtained using L pulses and $h_\gamma(n)$ is the impulse response of $H(z/\gamma) \equiv W(z)H(z)$. Computation of x_L and k_L is based on the minimization of

$$J(x_L, k_L) = \sum_{n=0}^{N-1} \left(e_W^{L+1}(n)\right)^2 \tag{1.22}$$

Setting $\frac{\partial J(x_L,k_L)}{\partial x_L} = 0$ yields

$$x_L \;=\; \frac{r_{eh}(k_L)}{r_{hh}(0)} \tag{1.23}$$

where $r_{eh}(m) \equiv \sum_n e_W^L(n)h_\gamma(n+m)$ and $r_{hh}(m) \equiv \sum_n h_\gamma(n)h_\gamma(n+m)$. By substituting expression (1.23) in (1.21) and the result into (1.22) we obtain

$$J(x_L, k_L)|_{x_L=\text{fixed}} \;=\; \sum_n (e_W^L(n))^2 - \frac{r_{eh}^2(k_L)}{r_{hh}(0)} \tag{1.24}$$

Thus, k_L is chosen so that $r_{eh}^2(k_L)$ in the above expression is maximized. The selected value of the location k_L is next used in (1.23) in order to compute the corresponding amplitude.

Recent extensions of the MPE method incorporate a long term prediction filter as well, activated when the speech segment is identified as voiced. The associated pitch period p in eq. (1.16) is determined by finding the first dominant coefficient of the autocorrelation $r_{ee}(m)$ of the unweighted residual, while the coefficient a_p is computed as

$$a_p = \frac{r_{ee}(p)}{r_{ee}(0)}\,. \tag{1.25}$$

1.4.3.2 Regular Pulse Excitation coding - RPE Regular Pulse Excitation methods are very similar to Multipulse Excitation ones. The basic difference is that the excitation sequence is of the form

$$x(n) = x_0\delta(n-k) + x_1\delta(n-k-p) + \ldots + x_{K-1}\delta(n-k-(K-1)p)\,, \tag{1.26}$$

i.e., impulses are equally spaced with a period p starting from the location k of the first impulse. Hence, the encoder should optimally select the initial impulse lag k, the period p and the amplitudes x_i ($i = 0, \ldots, K-1$) of all K impulses.

In its original form, proposed by Kroon et al. in [15] the encoder contains only a short term predictor of the form (1.18) and a perceptually weighting filter of the form (1.20). The steps followed by the RPE encoder are summarized next:

Pitch Estimation: The period p of the involved excitation sequence corresponds to the pitch period in the case of voiced segments. Hence an estimate of p can be obtained by inspecting the local maxima of the autocorrelation function of $s(n)$ as explained in Section 1.3.3.

Linear Prediction: The coefficients of $A_S(z)$ of the model in (1.18) are computed employing Linear Prediction (ref. end of Section 1.3.3).

Impulse Lag and Amplitude Estimation: This is the core step of RPE. The unknown lag k (i.e., the location of the first impulse) and all amplitudes x_i ($i = 0, \ldots, K-1$) are jointly estimated. Suppose that the $K \times 1$ vector $\mathbf{x}$ contains all x_i's. Then any excitation sequence $x(n)$ ($n = 0, \ldots, N-1$) with initial lag k can be written as an $N \times 1$ sparse vector, $\mathbf{x}^k$ with non-zero elements x_i located at k, $k+p$, $k+2p, \ldots, k+(K-1)p$. Equivalently,

$$\mathbf{x}^k = \mathbf{M}^k \mathbf{x} \tag{1.27}$$

where rows $k+ip$ ($i = 0, \ldots K-1$) of the $N \times K$ sparse binary matrix $\mathbf{M}$ contain a single 1 at their i-th position.

The perceptually weighted error attained by selecting a particular excitation $x(n)$ is

$$\begin{aligned} e(n) &= w(n) \star (s(n) - h(n) \star x(n)) \\ &= w(n) \star s(n) - h_\gamma(n) \star x(n) \end{aligned} \tag{1.28}$$

where $h(n)$ is the impulse response of the short term predictor $H_S(z)$, $h_\gamma(n)$ the impulse response of the cascade $W(z)H(z)$ and $s(n)$ the input speech signal. Equation (1.28) can be rewritten using vector notation as

$$\mathbf{e}^k = \mathbf{s}_w - \mathbf{H}_\gamma \mathbf{M}^k \mathbf{x} \tag{1.29}$$

where $\mathbf{s}_w$ is a $N \times 1$ vector depending upon $s(n)$ and the previous state of the filters that does not depend on k or $\mathbf{x}$ and $\mathbf{H}$ is an $N \times N$ matrix formed by shifted versions of the impulse response of $H(z/\gamma)$. The influence of k and x_i is incorporated in $\mathbf{M}^k$ and $\mathbf{x}$ respectively (see above for their definition).

For fixed k optimal $\mathbf{x}$ is the one that minimizes

$$\sum_{n=0}^{N-1} e(n)^2 = \left(\mathbf{e}^k\right)^T \left(\mathbf{e}^k\right), \tag{1.30}$$

that is

$$\mathbf{x} = \left[\left(\mathbf{M}^k\right)^T \mathbf{H}_\gamma^T \mathbf{H}_\gamma \mathbf{M}^k\right]^{-1} \left(\mathbf{M}^k\right)^T \mathbf{H}_\gamma^T. \tag{1.31}$$

After finding the optimal $\mathbf{x}$ for all candidate values of k using the above expression the overall optimal combination (k, x) is the one that yields the minimum squared error in eq. (1.30). Although the computational load due to matrix inversion in expression (1.31) seems to be extremely high, the internal structure of the involved matrices allows for fast implementations.

RPE architecture described above contains only a short term predictor $H_S(z)$. The addition of a long term predictor $H_L(z)$ of the form (1.16) enhances coding performance for high pitch voiced speech segments. Computation of the pitch period p and

the coefficient a is carried out by repetitive recalculation of the attained weighted MSE for various choices of p.

1.4.3.3 Code Excited Linear Prediction coding - CELP CELP is the most distinguished representative of Analysis-by-Synthesis codecs family. It was originally proposed by M.R. Schroeder and B.S. Atal in [16]. This original version of CELP employs both long and short term synthesis filters and its main innovation relies on the structure of the excitation sequences used as input to these filters. A collection of predefined pseudo-*Gaussian* sequences (vectors) of 40 samples each form the so called *Codebook* available both to the encoder and the decoder. A codebook of 1024 such sequences is proposed in [16].

Incoming speech is segmented into frames of The encoder performs a sequential search of the codebook in order to find the code vector that produces the minimum error between the synthetically produced speech and the original speech segment. In more detail, each sequence v_k ($k = 0, \ldots, 1023$) is multiplied by a gain g and passed to the cascade of the two synthesis filters (LTP and STP). The output is next modified by a perceptually weighting filter $W(z)$ and compared against an also perceptually weighted version of the input speech segment. Minimization of the resulting MSE allows for estimating the optimal gain for each code vector and finally for selecting that code vector with the overall minimum perceptual error.

The parameters of the short term filter ($H_S(z)$) that has the common structure of eq. (1.18) are computed using standard linear prediction optimization once for each frame, while long term filter ($H_L(z)$) parameters, i.e., p and a are recomputed within each sub-frame of 40 samples. In fact, a range $[20, \ldots, 147]$ of integer values of p are examined assuming no excitation. Under this assumption the output of the LTP depends only on past (already available) values of it (ref. eq. (1.17)). The value of a that minimizes perceptual error is computed for all admissible p's and the final value of p is the one that yields the overall minimum.

The involved perceptual filter $W(z)$ is constructed dynamically as function of $H_L(z)$ in a fashion similar to MPE and LPE.

The encoder transmits (a) quantized expressions of the LTP and STP coefficients, (b) the index k of the best fitting codeword, (c) the quantized version of the optimal gain g.

The decoder resynthesizes speech exciting the reconstructed copies of LTP and STP filters by the code vector k.

The descent quality of CELP encoded speech even at low bitrates captured the interest of the scientific community and the standardization bodies as well. Major research goals included (i) complexity reduction especially for the codebook search part of the algorithm and (ii) improvements on the delay introduced by the encoder. This effort resulted in a series of variants of CELP like VSELP, LD-CELP and ACELP that are briefly presented in the sequel.

Vector-Sum Excited Linear Prediction (VSELP): This algorithm was proposed by Gerson and Jasiuk in [17] and offers faster codebook search and improved robustness to possible transmission errors.

VSELP assumes three different codebooks; three different excitation sequences are extracted from them, multiplied by their own gains and summed up to form the input to the short term prediction filter. Two of the codebooks are static each of them containing 128 predifined pseudo-random sequences of length 40. In fact, each of the 128 sequences corresponds to a linear combination of 7 basis vectors weighted by ± 1.

On the other hand the third codebook is dynamically updated to contain the state of the autoregressive LTP $H_L(z)$ of eq. (1.16). Essentially, the sequence obtained from this adaptive codebook are equivalent to the output of the LTP filter for a particular choice of the lag p and the coefficient a. Optimal selection of p is performed in two stages: an open-loop procedure exploits the autocorrelation of the original speech segment, $s(n)$, to obtain a rough initial estimate of p. Then a closed-loop search is performed around this initial lag value to find this combination of p and a that at the absence of other excitation (from the other two codeebooks) produces synthetic speech as close to $s(n)$ as possible.

Low Delay CELP (LD-CELP): This version of CELP is due to J-H. Chen et. al. [18]. It applies very fine speech signal partitioning into frames of only 2.5 ms consisting of 4 subframes of 0.625 msec. The algorithm does not assume long term prediction (LTP) and employs 50 order short term prediction (STP) filter whose coefficients are updated every 2.5 msec. Linear prediction uses a novel autocorrelation estimator using only integer arithmetic.

Algebraic CELP (ACELP): ACELP has all the characteristics of the original CELP with major difference the simpler structure of its codebook. The latter contains ternary valued sequences, $c(n)$, ($c(n) \in \{-1, 0, 1\}$), of the form

$$c(n) = \sum_{i=1}^{K} (\alpha_i \delta(n - p_i) + \beta_i \delta(n - q_i)) \tag{1.32}$$

where $\alpha_i, \beta_i = \pm 1$, typically $K = 2, 3, 4$ or 5 (depending on the target bitrate) and the pulse locations p_i, q_i have a small number of admissible values. Table 1.1 includes these values for $K = 5$. This algebraic description of the code vectors allows for compact encoding and also for fast search within the codebook.

Relaxation Code Excited Linear Prediction coding (RCELP): RCELP algorithm [19], deviates from CELP in that it does not attempt to match the pitch of the original signal, $s(n)$, exactly. Instead, pitch is estimated once within each frame and linear interpolation is used for approximating the pitch in the intermediate time points. This reduces the number of bits used for encoding of pitch values.

$p_1,\ q_1 \in$	$\{0,$	$5,$	$10,$	$15,$	$20,$	$25,$	$30,$	$35\}$
$p_2,\ q_2 \in$	$\{1,$	$6,$	$11,$	$16,$	$21,$	$26,$	$31,$	$36\}$
$p_3,\ q_3 \in$	$\{2,$	$7,$	$12,$	$17,$	$22,$	$27,$	$32,$	$37\}$
$p_4,\ q_4 \in$	$\{3,$	$8,$	$13,$	$18,$	$23,$	$28,$	$33,$	$38\}$
$p_5,\ q_5 \in$	$\{4,$	$9,$	$14,$	$19,$	$24,$	$29,$	$34,$	$39\}$

Table 1.1

1.5 SPEECH CODING STANDARDS

Speech coding standards applicable to wireless communications are briefly presented in this section.

ITU G.722.2 (see [20]) Specifies wide-band coding of speech at around 16 kbit/s using the so called Adaptive Multi-Rate Wideband (AMR-WB) codec. The latter is based on ACELP. The standard describes encoding options targeting bitrates from 6.6 to 23.85 Kbps. The entire codec is compatible to the AMR-WB codecs of ETSI-GSM and 3GPP (specification TS 26.190).

ITU G.723.1 (see [21]) Uses Multi-Pulse Maximum Likelihood Quantization (MP-MLQ) and the ACELP speech codec. Target bitrates are 6.3 Kbps and 5.3 Kbps respectively. The coder operates on 30 msec frames of speech sampled at an 8 kHz rate.

ITU G.726 (see [22]) This specification refers to the conversion of linear or A-law or μ-law PCM to and from a 40, 32, 24 or 16 Kbps bitstream. Some ADPCM coding scheme is used.

ITU G.728 (see [23]) Uses LD-CELP to encode speech sampled at $8,000$ samples/sec with 16 Kbps.

ITU G.729 (see [24]) Specifies the use of Conjugate Structure ACELP algorithm for encoding speech at 8 Kbps.

ETSI-GSM 06.10 (see [25]) Specifies *GSM Full Rate (GSM-FR)* codec that employs RPE algorithm for encoding speech sampled at $8,000$ samples/sec. Target bitrate is 12.2 Kbps, i.e., equal to the throughput of GSM Full Rate channels.

ETSI-GSM 06.20 (see [26]) Specifies *GSM Half Rate (GSM-HR)* codec that employs VSELP algorithm for encoding speech sampled at $8,000$ samples/sec. Target bitrate is 5.6 Kbps, i.e., equal to the throughput of GSM Half Rate channels.

ETSI-GSM 06.60 (see [27]) Specifies *GSM Enhanced Full Rate (GSM-EFR)* codec that employs Conjugate Structure ACELP (CS-ACELP) algorithm for encoding speech sampled at $8,000$ samples/sec. Target bitrate is 12.2 Kbps, i.e., equal to the throughput of GSM Full Rate channels.

ETSI-GSM 06.90 (see [28]) Specifies *GSM Adaptive Multi-Rate (GSM-AMR)* codec that employs Conjugate Structure ACELP (CS-ACELP) algorithm for encoding speech sampled at $8,000$ samples/sec. Various target bitrate modes are supported starting from 4.75 Kbps up to 12.2 Kbps. A newer version of GSM-AMR, GSM WideBand AMR, was adopted by ETSI/GSM for encoding wideband speech sampled at $16,000$ samples/sec.

3GPP2 EVRC Adopted by the 3GPP2 consortium (under ARIB: STD-T64-C.S0014-0, TIA: IS-127 and TTA: TTAE.3G-C.S0014), specifies the so called *Enhanced Variable Rate Codec (EVRC)* that is based on RCELP speech coding algorithm. It supports three modes of operation targeting bitrates of 1.2, 4.8 and 9.6 Kbps.

3GPP2 SMV Adopted by 3GPP2 (under TIA: TIA-893-1) specifies the *Selectable Mode Vocoder (SMV)* for Wideband Spread Spectrum Communication Systems. SMV is CELP based and supports four modes of operation targeting bitrates of 1.2, 2.4, 4.8 and 9.6 Kbps.

1.6 UNDERSTANDING VIDEO CHARACTERISTICS

1.6.1 Video Perception

Color information of a point light source is represented by a 3×1 vector $\mathbf{c}$. This representation is possible due to human visual perception mechanism. In particular, color sense is a combination of the stimulation of three different types of *cones* (light sensitive cells spread on the retina). Each cone type has different frequency response when it is excited by the visible light (with wavelength $\lambda \in [\lambda_{min},\ \lambda_{max}]$ where $\lambda_{min} \approx 360nm$ and $\lambda_{max} \approx 830nm$). For a light source with spectrum $f(\lambda)$ the produced stimulus reaching the vision center of the brain is equivalent to the the vector,

$$\mathbf{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}, \quad \text{where } \ c_i = \int_{\lambda_{min}}^{\lambda_{max}} s_i(\lambda) f(\lambda) d\lambda, \ \ i = 1, 2, 3. \tag{1.33}$$

Functions $s_i(\lambda)$ attain their maxima in the neighborhoods of Red (R), Green (G) and Blue (B) as illustrated in Figure 1.8.

1.6.2 Discrete Representation of Video - Digital Video

Digital Video is essentially a *sequence* of still images of fixed size,i.e.,

$$\mathbf{x}(n_c, n_r, n_t),\ n_c = 0, \ldots, N_c - 1,\ n_r = 0, \ldots, N_r - 1,\ n_t = 0,\ 1,\ \ldots \quad (1.34)$$

where N_c, N_r are the numbers of columns and rows of each single image in the sequence and n_t determines the order of the particular image with respect to the very first one. In fact, if T_s is the time interval between capturing or displaying two successive images of the above sequence, $T_s n_t$ is the time elapsed between the acquisition/presentation of the first image and the n_t-th one.

The feeling of smooth motion requires presentation of successive images at rates higher than 10 to 15 per second. Almost perfect sense of smooth motion is attained using 50 to 60 changes per second. The latter correspond to $T_s = 1/50$ or $1/60$ sec.

Considering for example the European standard PAL for the representation of digital video $N_r = 576$, $N_c = 720$ and $T = 1/50$ sec. Simple calculations indicate that an overwhelming amount of approximately 20×10^6 samples should be captured/displayed per second. This raises the main issue of digital video handling: extreme volumes of data. The following sections are devoted to how these volumes can be represented in compact ways particularly for video transmission purposes.

1.6.2.1 Color Represantation In the previous paragraph we introduced the representation $\mathbf{x}(n_c, n_r, n_t)$ associating it with the somehow vague notion of video sample. Indeed digital video is a 3-D sequence of samples, i.e., measurements and more precisely measurements of color. Each of these samples is essentially a vector

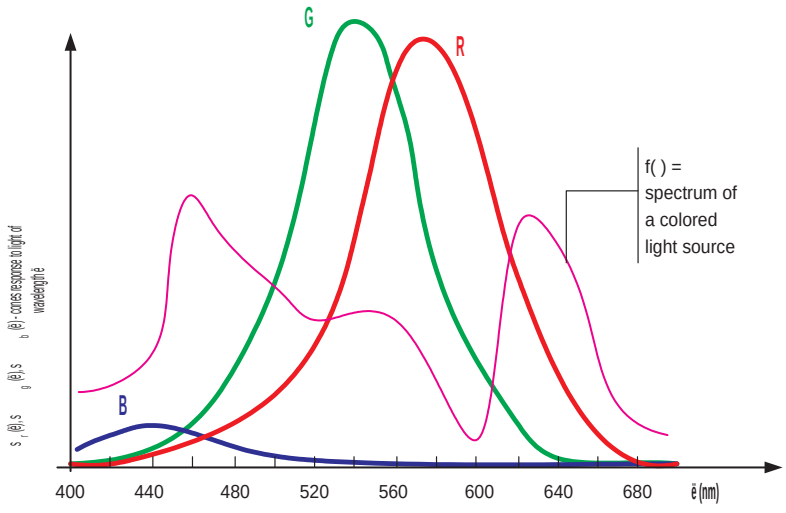

Fig. 1.8 Tri-stimulus response to colored light

usually of length 3 corresponding to a transformation

$$\mathbf{x} = \mathbf{T}\mathbf{c} \tag{1.35}$$

of the color vector of eq. (1.33).

RGB representation: In the simplest case

$$\mathbf{x} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{1.36}$$

where r, g, b are normalized versions of c_1, c_2 and c_3 respectively as defined in (1.33). In fact, since digital video is captured by video cameras rather the human eye the exact shape of $s_i(\lambda)$ in (1.33) depends on the particular frequency response of the acquisition sensors (e.g. CCD cells). Still though they are frequency selective and concentrated around the frequencies of Red, Green and Blue light. RGB representation is popular in the computer world but not that useful in video encoding / transmission applications.

YCrCb representation: Preferred color representation domain for video codecs is YCrCb. Historically this choice was due to compatibility constrains imposed moving from black and white television to color television; in this transition representing luminance (Y) as a discrete component and transmitting color information (Cr and Cb) through an additional channel provided backward compatibility. Digital video encoding and transmission stemming from their analog antecedents favor representations that decouple *luminance* from color. YCrCb is related to RGB through the transformation

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.500 & -0.4187 & -0.0813 \\ -0.1687 & -0.3313 & 0.500 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{1.37}$$

where Y represents the luminance level and Cr, Cb carry the information of color. Other types of transformation result to alternative representations like the YUV and YIQ that also contain a separate luminance component.

1.6.3 Basic Video Compression Ideas

1.6.3.1 Controlled Distortion of Video

Frame Size Adaptation: Depending on the target application video frame size varies from as low as 96×128 samples per frame for low quality multimedia presentations up to 1080×1920 samples for high definition television. In fact, video frames for

digital cinema reach even higher sizes. Second column of Table 1.2 includes standard frame sizes of most popular video formats.

Most cameras capture video in either PAL (in Europe) or NTSC (in the U.S.) and subsampling to smaller frame sizes is performed prior to video compression.

The typical procedure for frame size reduction contains the following steps:

1. Abortion of odd lines
2. Application of horizontal decimation i.e., low pass filtering and $2:1$ subsampling.

Format	Size	Framerate (fps)	Interlaced	Color Representation
CIF	288×352		NO	4:2:0
QCIF	144×176		NO	4:2:0
SQCIF	96×128		NO	4:2:0
SIF-625	288×352	25	NO	4:2:0
SIF-525	240×352	30	NO	4:2:0
PAL	576×720	25	YES	4:2:2
NTSC	486×720	29.97	YES	4:2:2
HDTV	720×1280	59.94	NO	4:2:0
HDTV	1080×1920	29.97	YES	4:2:0

Table 1.2 Characteristics of common standardized video formats.

Frame Rate Adaptation: Human vision system (eye retina cells, nerves and brain vision center) acts as a low pass filter regarding the temporal changes of the captured visual content. A side effect of this *incapability* is that by presenting to our vision system sequences of still images every $50-60$ times per second is enough for generating the sense of smooth scene change. This fundamental observation is behind the idea of approximating moving images by sequences of still frames. Well before the appearance of digital video technology the same idea was (and is still) used in traditional cinema.

Thus, using frame rates in the range of $50-60$ fps yields a satisfactory visual quality. In certain bitrate critical applications such as video conference frame rates as low as $10-15$ fps are used, leading to obvious degradation of the achieved quality.

In fact, psycho-visual experiments led to halving the $50-60$ fps rates using an approach that cheats human vision system: The so called *interlaced* frames are split into even and odd *fields* containing the even and odd numbered rows of samples of the original frame. Altering successively the content of only the even or the odd fields $50-60$ times per second yields a satisfactory smoothness although this corresponds to actual frame rate of only $25-30$ fps.

Third column of Table 1.2 lists the standardized frame rates of the included popular video formats. Missing framerates are not subject to standardization. In addition,

fourth column of the table declares whether the corresponding video frames are interlaced.

Color Subsampling of video: Apart of backwards compatibility constraints that forced the use of YCrCb color representation for video codecs an additional advantage of this representation has been identified. Psychovisual experiments showed that human vision system is more sensitive to high spatial frequencies of luminance than in the same range of spatial frequencies of color components Cr and Cb. This allowed for subsampling Cr and Cb (i.e., using less chrominance samples per frame) without serious visible deterioration of the visual content. Three main types of this color subsampling have been standardized: (i) *4:4:4* where no color subsampling is performed, (ii)*4:2:2* where for every 4 samples of Y only 2 samples of Cr and 2 samples of Cb are encoded, (iii) *4:2:0* where for every 4 samples of Y only 1 sample of Cr and 1 sample of Cb is encoded. The last column of Table 1.2 refers to the color subsampling scheme used in the included video formats.

Accuracy of color representation: Usually both luminance (Y) and color (Cr and Cb) samples are quantized to 2^8 levels and thus 8 bits are used for their representation.

1.6.3.2 Redundancy Reduction of Video

Motion Estimation and Compensation: Motion Estimation aims in reducing temporal correlation between successive frames of a video sequence. It is a technique analogous to prediction used in DPCM and ADPCM. Motion estimation is applied to selected frames of the video sequence in the following way:

1. *Macroblock grouping*:
   Pixels of each frame are grouped into macroblocks usually consisting of $4\,8 \times 8$ luminance (Y) blocks and from a single 8×8 block for each chrominance component (Cr and Cb for the YCrCb coloor representation). In fact this grouping is compatible to 4:2:0 color subsamplinng scheme. If 4:4:4 or 4:2:2 is used grouping is modified accordingly.

2. *Motion estimation:*
   Motion estimation for the macroblocks of a frame corresponding to current time index n a past or future frame corresponding to time m is used as a reference. For each macroblock, say B_n, of the current frame a search procedure is employed to find some 16×16 region, say M_m, of the reference frame whose luminance best matches the 16×16 luminance samples of B_n. Matching is evaluated on the basis of some distance measure such as the sum of the squared differences or the sum of the absolute differences between the corresponding luminance samples.

   The outcome of motion estimation is a *motion vector* for every macrobloock i.e., a 2×1 vector, $\mathbf{v}$ equal to the relative displacement between B_n and M_m.

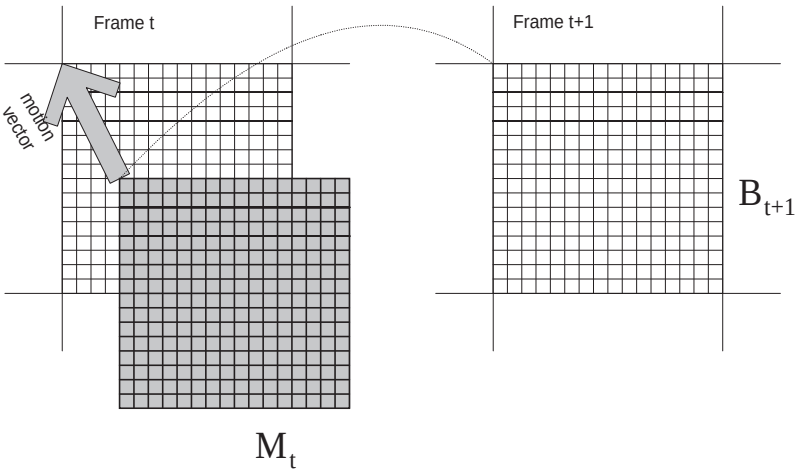

Fig. 1.9 Motion estimation of a 16×16 macroblock, B_{t+1} of $t+1$-frame using t-frame as a reference. In this example the resulting motion vector is $\mathbf{v} = (-4,\ 8)$.

3. *Calculation of Motion Compensated Prediction Error (Residual):*
   In the sequel instead of coding the pixel values of each macroblock, the difference macroblock
   $$E_n \stackrel{\triangle}{=} B_n - M_m$$
   is computed and coded. The corresponding motion vector is encoded as well.
   Figure 1.9 illustrates the aforementioned procedure.

Every few frames motion estimation is interrupted and particular frames are encoded rather than their motion compensated residuals. This prevents accumulation of errors and offers access points for restarting decoding. In general video frames are categorized into three types: I, P, and B.

- Type I or *Intra* frames are those that are independently encoded. No motion estimation is performed for the blocks of this type of frames.
- Type P or *Predicted* frames are those that are motion compensated using as reference the most recent of the *past* Intra or Predicted frames. Time index $n > m$ in this case for P frames.
- Type B or *Bidirectionnaly Interpolated* frames are those that are motion compensated with reference to past *and/or future* I and P frames. Motion estimation results in this case to two different motion vectors one for each of the past and future reference frames pointing to best matching regions M_{m^-} and M_{m^+} respectively. Motion error macroblock (that is passed to the next coding stages) is computed as
  $$E_n \stackrel{\triangle}{=} B_n - \frac{1}{2}(M_{m^-} + M_{m^+})$$

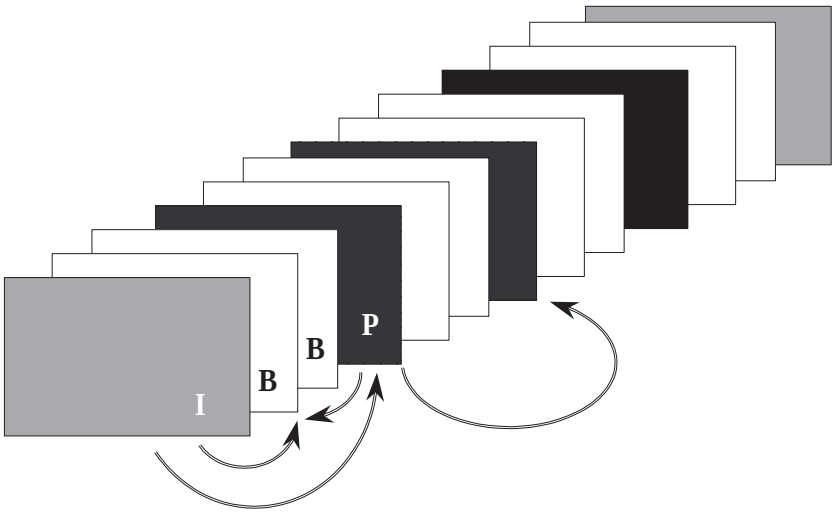

Fig. 1.10 Ordering and prediction reference of I, P and B frames within a GOP of 12 frames.

Usually video sequence is segmented into consecutive *Groups of Pictures (GOP)* starting with an I frame followed by type P and B frames located in predefined positions. GOP of Figure 1.10 has the structure IBBPBBPBBPBB. During decoding frames are reproduced in a different order since decoding of a B frame requires the subsequent I or P reference frames. For the previous example the following order should be used: I**BB**PBBPBBPBB**I**BB where the bold face B frames belong to the previous GOP and the bold face I frame belongs to the next GOP.

Transform Coding - The Discrete Cosine Transform (DCT): While motion estimation techniques are used to remove temporal correlation, DCT is used to remove spatial correlation. DCT is applied on 8×8 blocks of luminance and chrominance. The original sample values are transformed in the case of I frames and the prediction error blocks for P and B frames. If X represents any of these blocks the resulting transformed block, Y, is also 8×8 and is obtained as,

$$Y = F\, X \mathbb{F}^T \tag{1.38}$$

where the real valued 8×8 DCT transformation matrix F is of the form

$$F_{kl} = \begin{cases} \frac{1}{2} \cos\left(\frac{\pi}{8} k(l+\frac{1}{2})\right), & k = 1, \cdots 7 \\ \frac{1}{2\sqrt{2}} \cos\left(\frac{\pi}{8} k(l+\frac{1}{2})\right), & k = 0\,. \end{cases} \tag{1.39}$$

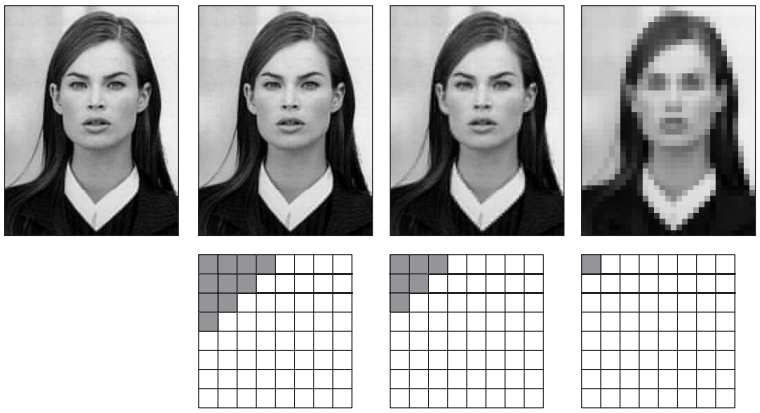

Fig. 1.11 Result of applying block DCT, discard least significant coefficients and the inverse DCT. The original image is the left-most one. Dark positions on the 8×8 grids of the lower part of the figure indicate the DCT coefficients that were retained before inverse DCT.

Decorrelation properties of DCT have been proved by extensive experiments on natural image data. Beyond decorrelation DCT exposes excellent energy compaction properties. In practice this means that most informative DCT coefficients within Y are positioned close to the upper-left portion of the transformed block. This behavior is demonstrated in Figure 1.11 where a natural image (top - left) is transformed using block DCT, least significant coefficients were discarded (set to 0) and an approximation of the original image was produced by inverse DCT.

Apart from its excellent decorrelation and energy compaction properties DCT is preferred in coding applications because a number of *fast DCT* implementations (some of them in hardware) are available.

1.7 VIDEO COMPRESSION STANDARDS

1.7.1 H.261

ITU video encoding international standard H.261 [29] was developed for use in video-conferencing applications over ISDN channels allowing bitrates of $p \times 64$ kbps, $p = 1 \cdots 30$. In order to bridge the gap between the European PAL and the North American NTSC video formats, H.261 adopts the Common Interchange Format (CIF) and for lower bitrates the QCIF (see Table 1.2). It is interesting to notice that even using QCIF / 4:2:0 with a framerate of 10 frames per second requires a bitrate of approximately 3 Mbps which means that a compression ratio of $48 : 1$ is required in order to transmit it over a 64 Kbps ISDN channel.

H.261 defines a hierarchical data structure. Each frame consists of 12 GOB (group-of-blocks). Each GOB contains 33 MacroBlocks (MB) that are further split into 8×8

Blocks (B). Encoding parameters are assumed unchanged within each macroblock. Each MB consists of 4 Luminance (Y) blocks 2 chrominance blocks in accordance to 4:2:0 color subsampling scheme.

The standard adopts a hybrid encoding algorithm using Discrete Cosine Transform (DCT) and Motion Compensation between successive frames.

Two modes of operation are supported:

Interframe coding: In this mode already encoded (decoded) frames are buffered in the memory of the encoder (decoder). Motion estimation is applied on *Macroblocks* of the current frame using the previous frame as a reference (Type P frames). Motion compensation residual is computed by subtracting the best matching region of the previous frame from the current Macroblock. The six 8×8 Blocks of the residual are next DCT transformed. DCT coefficients are quantized and quantization symbols are entropy encoded. Non zero motion vectors are also encoded. The obtained bitstream containing (a) the encoded quantized DCT coefficients, (b) the encoded non-zero motion vectors and (c) the parameters of the employed quantizer is passed to the output buffer that guarantees constant outgoing bitrate. Monitoring the level of this same buffer a control mechanism determines the quantization quality for the next Macroblocks in order to avoid overflow or underflow.

Intraframe coding: In order (a) to avoid accumulation of errors, (b) to allow for (re)starting of the decoding procedure at arbitrary time instances and (b) for improving image quality in the case of abrupt changes of video content (where motion compensated prediction fails to offer good estimates) the encoder supports block DCT encoding of selected frames (instead of motion compensation residuals). These Type I frames may appear in arbitrary time instances; it is a matter of the particular implementation to decide when and under which conditions an Intra frame will be inserted.

Either Intra blocks or motion compensation residuals are DCT transformed and quantized.

H.261 decoder follows in a straightforward manner the inverse procedure.

1.7.2 H.263

H.263 ITU standard [30] is descendant of H.261 offering better encoding quality especially for low bitrate applications that are its main target. In comparison to H.261 it incorporates more accurate motion estimation procedures resulting in motion vectors of half-pixel accuracy. In addition, motion estimation can switch between 16×16 and 8×8 block matching; this offers better performance especially in high detail image areas. H.263 supports bi-directional motion estimation (B frames) and the use of arithmetic coding of the DCT coefficients.

1.7.3 MPEG-1

MPEG-1 ISO standard [10] produced by the Motion Pictures Expert Group is the first in a series of video (and audio) standards produced by this group of ISO. In fact, the standard itself describes the structure and the semantics of the encoded stream in order to be decodable by an MPEG-1 compliant decoder. The exact operation of the encoder and the employed algorithms (e.g. motion estimation search method) are on purpose left as open design issues to be decided by developers in a competitive manner.

MPEG-1 is targeting video and audio encoding at bitrates in the range of 1.5 Mbits/sec. Approximately 1.25 Mbits/sec are assigned for encoding SIF-625 or SIF-525 non-interlaced video and 250 Kbits/sec for stereo audio encoding. MPEG-1 was originally designed for storing/playing back video to/from single speed CD-ROMs.

The standard assumes a CCIR 601 input image sequence i.e., images with 576 lines with 720 luminance samples and from 360 samples for Cr and Cb. Incoming frame rate is up to 50 fps.

Input frames are lowpass filtered and decimated to 288 lines of 360 (180) luminance (chrominance) samples.

Video codec part of the standard relies on the use of

- Decimation for downsizing the original frames to SIF. Interpolation at the decoder's side.
- Motion estimation and compensation as described in Section 1.6.3.2.
- Block DCT on 8×8 blocks of luminance and chrominance.
- Quantization of the DCT coefficients using a dead zone quantizer. Appropriate amplification of the DCT coefficients prior to quantization results in finer resolution for the most significant of them and suppression of the weak high-frequency ones.
- Run Length Encoding (RLE) using zig-zag scanning of the DCT coefficients (see Figure 1.12). In particular, if $s(0)$ is the symbol assigned to the dead zone (to DCT coefficients around zero) and $s(i)$ any other quantization symbol, RLE represents symbol strings of the form

$$\underbrace{s(0) \ \cdots \ s(0)}_{n} s(i), \quad n \geqslant 0,$$

  with new shortcut symbols A_{ni} indicating that n zeros ($s(0)$)are followed by the symbol $s(i)$.
- Entropy coding of the *run* symbols A_{ni} using Huffman Variable Length Encoder.

MPEG-1 encoders support bitrate control mechanisms. The produced bitstream is passed to control FIFO buffer that empties with a rate equal to the *target bitrate*. When

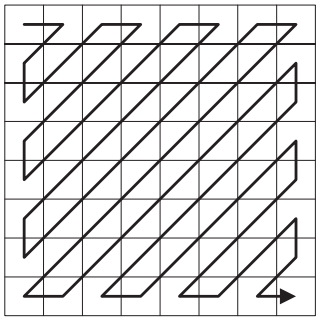

Fig. 1.12 Reordering of the 8×8 quantized DCT coefficients into a 64×1 linear array uses the zig-zag convention. Coefficients corresponding to DC and low spatial frequencies are scanned first while highest frequencies are left for the very end of the array. Normally this results in long zero-runs after the first few elements of latter.

the level of the buffer exceeds a predefined threshold the encoder forces its quantizer to reduce quantization quality (e.g., increase the width of the dead zone) leading of course to deterioration of encoding quality as well. On the opposite, when the control buffer level becomes lower than a certain bound the encoder forces finer quantization of the DCT coefficients. This mechanism guarantees an actual average bitrate very close to target bitrate. This type of encoding is known as Constant Bit Rate (CBR). When the control mechanism is absent or activated only in very extreme situations encoding quality remains almost constant but the bitrate is strongly changing leading to the so-called Variable Bitrate (VBR) encoding.

1.7.4 MPEG-2

MPEG-2 ISO standard [31] has been designed for high bitrate applications typically starting from 2 and reaching up to 60 Mbps. It can handle a multitude of input formats including CCIR-601, HDTV, 4K etc. Contrary to MPEG-1, MPEG-2 allows for interlaced video, very common in broadcast video applications, and exploits the redundancy between odd and even fields. Target uses of the standard include digital television, high definition television, DVD and digital cinema.

The core encoding strategies of MPEG-2 are very close to those of MPEG-1; perhaps its most important improvement relies on the so called *scalable coding* approach. Information scaling refers to the subdivision of the encoded stream into separate sub-streams that carry different levels of information detail. One of them transports the absolutely necessary information that is available to all users (decoders) while the others contain complementary data that may upgrade the quality of the received video. Four types of scalability are supported:

- *Data Partitioning* where for example the basic stream contains information only for low frequency DCT coefficients while high frequency DCT coefficients can be retrieved only through the complementary streams.
- *SNR Scalability* where the basic stream contains information regarding the most significant bits of the DCT coefficients (equivalent to coarse quantization) while the other streams carry least significant bits information.
- *Spatial Scalability* where the basic stream encodes a low resolution version of the video sequence and complementary streams may be used for improving image analysis.
- *Temporal Scalability* where complementary streams encode time decimated versions of the video sequence while their combination may increase temporal resolution.

1.7.5 MPEG-4

Contrary to MPEG-1/2, that introduced particular compression schemes for audio-visual data, MPEG-4 ISO standard [32] concentrates on the combined management of versatile multimedia sources. Different codecs are supported within the standard for optimal compression of each of these sources. MPEG-4 adopts the notion of an audiovisual scene that is composed of multiple *Audiovisual Objects (AVO)* that evolve both in space and time. These objects may be

1. Moving images (natural video) or space/time segments of them
2. Synthetic (computer generated) video
3. Still images or segments of them
4. Synthetic 2-D or 3-D objects
5. Digital sound
6. Graphics or
7. Text

MPEG-4 encoders use existing encoding schemes (such as MPEG-1 or JPEG) for encoding the various types of audiovisual objects. Their most important task is to handle AVO hierarchy (e.g., the object *newscaster* comprises of the lower level objects: *moving image of the newscaster* and *voice of the newscaster*. Beyond that MPEG-4 encodes the time alignment of the encoded AVOs.

Major innovation of MPEG-4 is that it assigns the *Synthesis procedure* of the final form of the video to the end viewer. The viewer (actually the decoder parametrized by the viewer) receives the encoded information of the separate AVOs and is responsible for the final synthesis possibly in accordance to an instructions stream distributed by

the encoder. Instructions are expressed using an MPEG-4 specific language called Binary Format for Scenes - BIFS that is very close to VRML. A brief presentation of MPEG-4 can be found in [33] while a detailed description of BIFS is presented in [34].

The resulting advantages of MPEG-4 approach are summarized below:

1. It may offer *better compression* rates of natural video by adapting compression quality or other encoding parameters (like the motion estimation algorithm) to the visual or semantic importance of particular portions of it. For example, background objects can be encoded with less than the important objects of the foreground.

2. It provides a genuine handling of different multimedia modalities. For example, text or graphics need not be encoded as pixels rasters superimposed on video pixels; both of them can be encoded separately using their native codecs postponing superposition till the synthesis procedure at decoding stage.

3. It offers advanced levels of interaction since it assigns to the end user the task of (re-) synthesizing the transmitted audiovisual objects into an integrated scene. In fact, MPEG-4 players instead of being dummy decoders they can be *interactive multimedia applications*. For example, consider a scenario of sports game where together with the live video, MPEG-4 encoders streams game statistics in textual form, information regarding the participating players etc.

1.7.6 H.264

H.264 is the first video (and audio) compression standard [35] to be produced by the *combined standardization* efforts of ITU's Video Coding Experts Group (VCEG) and ISO's Motion Pictures Experts Group (MPEG). The standard, released in 2003, defines five different *profiles* and overall 15 *levels* distributed with these profile. Each profile determines *a subset* of the syntax used by H.264 to represent encoded data. This allows for adapting the complexity of the corresponding codecs to the actual needs of particular applications. For the same reason different levels within each profile limit the options for various parameter values (like the size of particular look-up tables). Profiles and levels are ordered according to the quality requirements of targeted applications. Indicatively, Level 1 (within Profile 1) is appropriate for encoding video with up to 64 Kbps. At the other end Level 5.1 within profile 5 is considered for video encoding at bitrates up to $240,000$ Kbps.

H.264 achieves much better video compression - two or three times lower bitrate for the same quality of decoded video - than all previous standards at the cost of increased coding complexity. It uses all tools described in Section 1.6.3 for controlled distortion, subsampling, redundancy reduction (via motion estimation) transform coding and entropy coding also used in MPEG-1/2 and H.261/3 with some major innovations. These innovative characteristics of H.264 are summarized below.

Intra Prediction: Motion estimation as presented within Section 1.6.3.2 was described as a means for reducing temporal correlation in the sense that macroblocks of a B or P frame, with time index n, are predicted from blocks of equal size (perhaps in different locations) of *previous or/and future* reference frames, of time index $m \neq n$. H.264 recognizes that a macroblock may be similar to another macroblock within the same frame. Hence motion estimation and compensation is extended to intra frame processing (reference coincide with current frame, $m = n$) searching for self similarities. Of course, this search is limited to portions of the same frame that will be available to decoder prior to the currently encoded macroblock. On top of that computation of the residual,

$$E_n \stackrel{\triangle}{=} B_n \; - \; \hat{M}_n$$

is based on a decoded version of the reference region. Using this approach H.264 achieves not only temporal decorrelation (with the conventional inter motion estimation) but also spatial decorrelation.

In addition, macroblocks are allowed to be non-square shaped and non-equally sized (apart of 16×16, sizes of 16×8, 8×16, 8×8 are allowed). In Profile 1, Macroblocks are further split into *blocks* of 4×4 luminance samples and 2×2 chrominance samples; Larger 8×8 blocks are allowed in higher profiles. This extra degree of freedom offers better motion estimation results especially for regions of high detail where matching fails for large sized macroblocks.

Integer Arithmetic Transform: H.264 introduces a deviation of Discrete Cosine Transform with integer valued transformation matrix $\mathbf{F}$ (ref. eq. (1.38)). In fact, entries of $\mathbf{F}$ assume slightly different values depending on whether the transform is applied on intra blocks or residual (motion compensated) blocks, luminance or chrominance blocks. Entries of $\mathbf{F}$ are chosen in a way that both the direct and inverse transform can be implemented using only bit-shifts and additions (multiplication free).

Improved Lossless Coding: Instead of the conventional Run Length Encoding (RLE) followed by Huffman or Arithmetic entropy coding, H.264 introduces two other techniques for encoding the transformed residual sample values, the motion vectors e.t.c., namely,

1. *Exponential Golomb Code* is used for encoding single parameter values while *Context-based Adaptive Variable Length Coding (CAVLC)* is introduced as an improvement of the conventional RLE.

2. *Context-based Adaptive Binary Arithmetic Coding (CABAC)* is introduced in place of Huffman or conventional Arithmetic coding.

References

1. C. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, pp. 10–21, 1949.

2. D. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the I.R.E.*, vol. 40, pp. 1098–1101, September 1952.

3. P. G. Howard and J. S. Vitter, "Analysis of arithmetic coding for data compression," *Information Processing and Management*, vol. 28, no. 6, pp. 749–764, 1992.

4. ITU-T, "Recommendation G.711 - pulse code modulation (PCM) of voice frequencies," *Geneva, Switzerland*, 1988.

5. B. Sklar, *Digital Communications: Fundamentals and Applications*. Englewood Cliffs, N.J.: Prentice-Hall, 1988.

6. S. Haykin, *Adaptive Filter Theory*. Upper Saddle River, NJ: Prentice-Hall, 3rd ed., 1996.

7. H. Nyquist, "Certain topics in telegraph transmission theory," *Trans. AIEE*, vol. 47, pp. 617–644, April 1928.

8. C. Lanciani and R. Schafer, "Psychoacoustically-based processing of MPEG-I Layer 1-2 encoded signals," 1997.

9. D. Pan, "A tutorial on MPEG/audio compression," *IEEE MultiMedia*, vol. 2, no. 2, pp. 60–74, 1995.

10. ISO/IEC, "MPEG-1 coding of moving pictures and associated audio for digital storage media at up to about 1,5 mbit/s," *ISO/IEC 11172*, 1993.

11. A. Spanias, "Speech coding: A tutorial review," *Proceedings of the IEEE*, vol. 82, pp. 1541–1582, October 1994.

12. R. M. B. Gold, P.E. Blankenship, "New applications of channel vocoders," *IEEE Trans. ASSP*, vol. 29, pp. 13–23, February 1981.

13. H. Dudley, "Remaking speech," *J. Acoust. Soc. Am.*, vol. 11, no. 2, pp. 169–177, 1939.

14. B. Atal and J. Remde, "A new model for LPC excitation for producing natural sound speech at low bit rates," *Proc.ICASSP-82*, vol. 1, pp. 614–617, May 1982.

15. E. D. P. Kroon and R. Sluyter, "Regular-pulse excitation: A novel approach to effective and efficient multi-pulse coding of speech," *IEEE Trans. ASSP*, vol. 34, pp. 1054–1063, October 1986.

16. M. Schroeder and B. Atal, "Code-excited linear prediction (CELP): High-quality speech at very low bit rates," *Proc. ICASSP*, pp. 937–940, March 1985.

17. I. Gerson and M. Jasiuk, "Vector sum excited linear prediction (VSELP) speech coding at 8 kbits/s," *Proc. ICASSP-90, New Mexico*, April 1990.

18. R. C. Y.-C. L. N. J. J-H., Chen and M. Melchner, "A low delay CELP coder for the CCITT 16 kbps speech coding standard," *IEEE J. Selected Areas in Communications*, vol. 10, pp. 830–849, June 1992.

19. P. K. W.B.Kleijn and D. Nahumi, "The RCELP speech-coding algorithm," *European Trans. on Telecommunications*, vol. 5, pp. 573–582, September/October 1994.

20. ITU-T, "Recommendation G.722.2 - wideband coding of speech at around 16 kbit/s using adaptive multi-rate wideband (AMR-WB)," *Geneva, Switzerland*, July 2003.

21. ITU-T, "Recommendation G.723.1 - dual rate speech coder for multimedia communications," *Geneva, Switzerland*, March 1996.

22. ITU-T, "Recommendation G.726 - 40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)," *Geneva, Switzerland*, December 1990.

23. ITU-T, "Recommendation G.728 - coding of speech at 16 kbit/s using low-delay code excited linear prediction," *Geneva, Switzerland*, September 1992.

24. ITU-T, "Recommendation G.729 - coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear-prediction (CS-ACELP)," *Geneva, Switzerland*, March 1996.

25. ETSI EN 300 961 V8.1.1, "GSM 6.10 - digital cellular telecommunications system (phase 2+); full rate speech; transcoding," *Sophia Antipolis Cedex - France*, November 2000.

26. ETSI EN 300 969 V8.0.1, "GSM 6.20 - digital cellular telecommunications system (phase 2+); half rate speech; half rate speech transcoding," *Sophia Antipolis Cedex - France*, November 2000.

27. ETSI EN 300 726 V8.0.1, "GSM 6.60 - digital cellular telecommunications system (phase 2+); enhanced full rate (EFR) speech transcoding," *Sophia Antipolis Cedex - France*, November 2000.

28. ETSI EN 300 704 V7.2.1, "GSM 6.90 - digital cellular telecommunications system (phase 2+); adaptive multi-rate (AMR) speech; transcoding," *Sophia Antipolis Cedex - France*, April 2000.

29. ITU-T, "Recommendation H.261 - video codec for audiovisual services at p x 64 kbit/s," *Geneva, Switzerland*, 1993.

30. ITU-T, "Recommendation H.263 - video coding for low bit rate communication," *Geneva, Switzerland*, February 1998.

31. ISO/IEC, "MPEG-2 generic coding of moving pictures and associated audio information," *ISO/IEC 13818*, 1996.

32. ISO/IEC, "Overview of the MPEG-4 standard," *ISO/IEC JTC1/SC29/WG11 N2323*, July 1998.

33. Rob Koenen, "MPEG-4 multimedia for our time," *IEEE Spectrum*, vol. 36, pp. 26–33, February 1999.

34. Julien Signes, "Binary Format for Scene (BIFS): Combining MPEG-4 media to build rich multimedia services," *SPIE Proceedings*, 1998.

35. ITU-T, "Recommendation H.264 - advanced video coding (avc) for generic audiovisual services," *Geneva, Switzerland*, May 2003.