Source Coding of Speech and Video Signals

SHARAD SINGHAL, MEMBER, IEEE, DIDIER LE GALL, MEMBER, IEEE, AND CHENG-TIE CHEN, MEMBER, IEEE

Invited Paper

We have seen rapid progress in source coding techniques for both speech and video over the last decade. Linear prediction, subband coding, transform coding, interframe prediction as well as vector quantization and improved entropy coding techniques have been used to design coding algorithms which can achieve substantially more compression than was thought possible only a few years ago. At the same time, advances in VLSI technology and high-speed ASICs have allowed cost-effective implementation of these algorithms in real-time systems.

Toll-quality speech is now possible at only 16 kbits/s and natural sounding speech can be obtained for as low as 4.8 kbits/s. Video can be coded at rates from 64 kbits/s for teleconferencing applications to 140 Mbits/s for high-definition television, with intermediate rates providing a range of quality and bit rates for different applications.

We review some digital source coding techniques for speech and video. We concentrate on those algorithms that offer high compression while maintaining the perceptual quality of the source signals. Although not all of the techniques discussed have matured enough to appear in communication systems, many have high potential for systems where channel bandwidth is limited.

I. Introduction

Coding techniques for both speech and video have made rapid progress in the last decade. Linear prediction, subband coding, transform coding, interframe prediction as well as vector quantization and improved entropy coding techniques have been used to design coding algorithms which can achieve substantially more compression than was thought possible only a few years ago.

Figure 1 shows a typical communications channel. The *source encoder* depends on the input signal and removes redundancy in the source signal. The output of one or more source encoders is combined at the *channel encoder* and sent over the communications channel after appropriate modulation. The channel encoder may also add framing and error recovery information to the signal. At the other end of the channel, the *channel decoder* demodulates the received signal, detects and/or corrects any channel errors, and separates the source signals. The *source decoder* then reproduces the original signal as faithfully as possible. Although combined source and channel encoding is possible and often desirable, in the interest of reduced complexity and design flexibility, the two are usually considered independently, i.e., the source coder and decoder depend only on the signal source without regard to the communication channel, and the channel coder and decoder usually do not depend on the signal source.

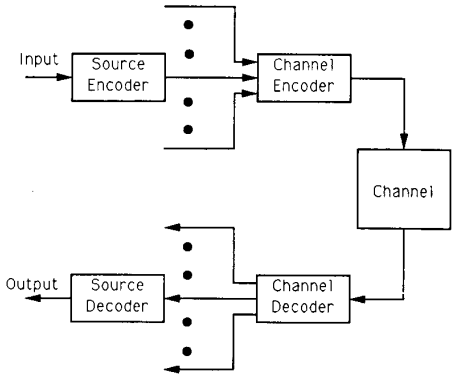


Fig. 1. Block diagram of a communications channel. The source encoder removes redundancy from the source signal. The channel encoder combines the output of one or more source coders for transmission over the channel. The channel decoder separates the source signals and the source decoder then reconstructs the input signal at its output.

In this paper, we review some digital source coding techniques for speech and video. We concentrate on those algorithms that maintain the perceptual quality of the source signals during compression. Our emphasis on perceptual quality leads us to include some algorithms that are general waveform coding algorithms and do not strictly depend on the input source, as well as exclude some traditional "source" coders which do not offer high quality. Although not all of the techniques described here have matured enough to appear in real communication systems, they have high potential for systems where the channel bandwidth is limited. We also restrict our treatment to source coding and do not discuss how these coders perform on different communication channels, specifically in the presence of channel errors. The coders are sensitive to bit errors in the coded bitstream because they remove some or most of the redundancy in the source signals. Usually, the perceptual quality of the output depends strongly on only a few parameters in the bitstream and error correction techniques can

Manuscript received March 14, 1990.

S. Singhal and C.-T. Chen are with Bell Communications Research, Morristown, NJ 07960, USA.

D. Le Gall was with Bell Communications Research. He is currently with C-Cube Microsystem, San Jose, CA 95131, USA.

IEEE Log Number 9036504.

0018-9219/90/0700-1233$01.00 © 1990 IEEE

be employed to give robust coders with only a small increase in bit rate.

II. Coding of Speech Signals

Although analog coding techniques are in widespread use for channel coding and video coding, source coders for speech signals invariably use digital techniques. Almost all speech coding techniques start from a digital representation of speech, obtained by band-limiting the analog signal and uniformly time-sampling as well as uniformly amplitude-quantizing it, using an analog-to-digital (A/D) converter. If the output codes from the A/D converter are labeled using the natural binary sequence, this representation is known as PCM (pulse code modulation). Since the input signal is band-limited, the original signal can be reconstructed without distortion from the corresponding (unquantized) time-sampled values provided the sampling rate is at least twice the signal bandwidth. Depending on the application, the bandwidth of the input speech may vary from about 3 kHz for telephony to nearly 20 kHz for high-fidelity audio with the corresponding sample rates varying from about 8 kHz to 48 kHz respectively. Unlike time-sampling, the amplitude quantization performed at the A/D converter inherently introduces distortion in the signal. If the quantization error is small, it can be assumed to be independent of the signal and a distortion measure in terms of the signal-to-noise ratio (SNR) can be defined. The SNR associated with uniform quantization is [1], [2]:

$$\mathrm{SNR}_{\mathrm{PCM}}(\mathrm{dB}) = 10 \log_{10}\mathrm{SNR}_{\mathrm{PCM}} \approx (6B - \theta) \qquad (1)$$

where B is the number of binary bits used in the representation and θ is a step-size dependent parameter. It should be noted that the above formulation requires that the quantization error be modeled as additive noise to the signal. For coarse quantization (say $B < 5$), the quantization error has too much structure and correlation with the input signal to be considered additive noise. Typically, about 12–13 bits are needed to assure adequate SNR for telephony and 16–18 bits are used for high-fidelity audio applications. For the purpose of this paper, we restrict our discussion to telephone bandwidth (200 Hz–3.4 kHz) speech and mention the higher bandwidth applications as applicable.

Equation (1) assumes that the range of the quantizer is aligned with the speech amplitudes at its input. Typical speech signals have a dynamic range of about 40 dB [2]. This implies that for low-amplitude portions of speech, the SNR is much lower than that computed from (1). This can be seen in Fig. 2 where the SNR is shown as a function of the signal amplitude. This problem can be mitigated somewhat by using *nonuniform* quantization. In principle, this is accomplished by a *compander*, which compresses the amplitude of the input signal s using a nonlinear characteristic $c(\cdot)$, uniformly quantizes the compressed signal, then expands the quantized version of the signal through the inverse compression characteristic $c^{-1}(\cdot)$. The most prevalent compression characteristics used for telecommunications are defined by the μ-law and A-law logarithmic companding [2] curves:

$$\mu\text{-law: } s_c = s_{\max} \frac{\ln (1 + \mu |s|/s_{\max})}{\ln (1 + \mu)} \operatorname{sgn}(s),$$

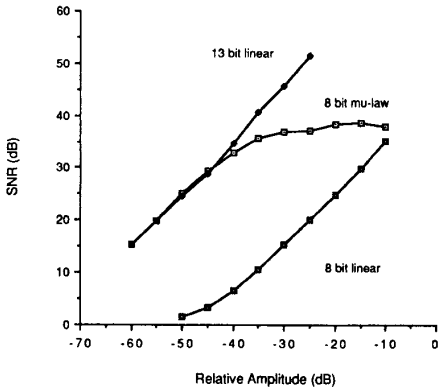


Fig. 2. The signal-to-quantization-noise ratio (SNR) as a function of the input amplitude for a sinusoidal input. The input amplitude is in dB relative to the input range of the quantizer.

$$A\text{-law: } s_c = \begin{cases} \dfrac{A|s|}{1 + \ln A} \operatorname{sgn}(s) & 0 \le \dfrac{|s|}{s_{\max}} \le \dfrac{1}{A}, \\ s_{\max} \dfrac{1 + \ln (A|s|/s_{\max})}{1 + \ln A} \operatorname{sgn}(s), & \\ & \dfrac{1}{A} \le \dfrac{|s|}{s_{\max}} \le 1, \end{cases} \qquad (2)$$

where s is a sample from the input speech and varies in the range $-s_{\max}$ to $s_{\max}$, and s_c is the compressed output. In practice, the input signal is quantized using 12- or 13-bit uniform quantizers and piecewise linear versions of the characteristics are used to convert the output codes digitally between the representations. Figure 2 also shows the SNR for an 8-bit μ-law coder as a function of the input amplitude. When compared to uniform quantization, it can be seen that the μ-law quantizer achieves a maximum SNR that is equivalent to 8-bit uniform quantization. However, unlike the uniform quantizer, the μ-law quantizer maintains this SNR for almost a 30-dB input range. Only about 7–8 bits suffice for toll-quality speech when using these quantizers, resulting in a bit rate of 56–64 kbits/s for speech sampled at 8 kHz. This quality forms an important point of reference when evaluating low bit-rate speech coders.

A. Predictive Coding of Speech

Although the speech signal is highly nonstationary in the long term, it shows substantial short-term stationarity. This implies that at time n, the current sample $s(n)$ of speech signal can be predicted from its recent history and the resulting difference $d(n)$ between $s(n)$ and its predicted value $\hat{s}(n)$ can be quantized for transmission over the channel. The difference signal $d(n)$ is also known as the *prediction residual*. At the receiver, the quantized prediction residual is integrated to reconstruct the output speech signal.

The simplest predictive coders are *differential PCM* or DPCM [2], [3] coders shown in Fig. 3. The predicted value $\hat{s}(n)$ of the current sample $s(n)$ is subtracted from $s(n)$ to obtain the difference signal $d(n)$ and quantized to $d'(n)$. The output of the decoder $s'(n) = d'(n) + \hat{s}(n)$ is used to predict the next input sample value. In general, linear predictors

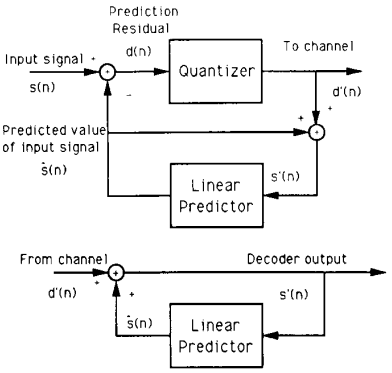


Fig. 3. Block diagram of a DPCM codec. The predicted value of the input signal is subtracted from the input to form the prediction residual which is quantized and sent to the decoder. The decoder adds back the predicted value to the received signal to form the output signal.

of the form

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s'(n-k) \tag{3}$$

are employed. The coefficients a_k are called predictor coefficients and are computed from the long-term statistics of speech signals. The feedback arrangement whereby the output of the decoder is used for prediction rather than the past input samples prevents accumulation of quantization errors at the decoder and, except for very coarse quantization, provides prediction gains comparable to the case where the prediction is based on the past input samples. Typically the prediction gain improves with filter order p up to order 3 or 4, then saturates [2], [3].

Further improvements are possible by allowing the predictor coefficients to adapt to the short-term statistics of the speech signal. The resulting class of coders are called *adaptive predictive coders* (APC) [2], [4]. As shown in Fig. 4, two

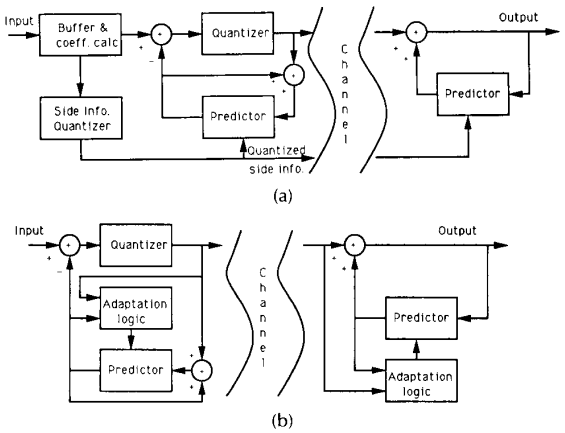


Fig. 4. Block diagrams of (a) forward and (b) backward adaptative predictive coders. The forward adaptive coder computes the predictor parameters from the incoming signal and transmits them as side information to the receiver. The backward adaptive coder computes the predictor coefficients from the quantized prediction residual at both the encoder and decoder.

different strategies are possible for predictor adaptation. The adaptation can be based on the incoming signal with the adaptation information explicitly transmitted to the receiver (forward adaptation), or it can be based on the quantized data with the receiver deriving the adaptation information from the quantized prediction residual (backward adaptation). With forward adaptation, the coder typically buffers speech in 10–20 ms frames, computes the predictor coefficients (and possibly the quantizer stepsize) and transmits these parameters as side information to the decoder as well as the quantized prediction residual. Backward adaptive coders compute the predictor (and quantizer parameters) from the quantized prediction residual and thus do not need the extra delay for buffering or the extra channel capacity for the side information. *Adaptive differential PCM* (ADPCM) [2], [5]–[7] coders using backward adaptive predictors and backward adaptive quantizers can achieve toll quality at roughly half the rate of μ-law PCM, and an ADPCM coder has been standardized for use at 32 kbits/s by the CCITT [5]. However, these coders suffer at lower bit rates because the prediction gain drops substantially as a result of coarse quantization of the prediction residual.

1) Computation of Adaptive Predictors: Backward adaptive coders have to compute the predictor coefficients using the quantized prediction residual or the reconstructed speech signal. Usually this is done by a number of algorithms based on the method of steepest descent or gradient search [8], [9]. An extremely simple solution to the problem is given by the least mean square (LMS) algorithm [2], [9]:

$$a_k(n+1) = a_k(n) + \alpha d'(n)s'(n-k), \quad 1 \le k \le p \tag{4}$$

where α is a small positive number much less than unity and $a_k(n)$ represents the kth parameter at time n. Modifications of this basic algorithm, such as using a nonunity multiplier on $a_k(n)$ term or making α dependent on n and/or k, allow for more robust and/or faster adaptation [9].

Forward adaptive coders assume that the predicted value of the current sample is

$$\hat{s}(n) = \sum_{k=1}^{p} a_k s(n-k). \tag{5}$$

The predictor coefficients, also known as LPC coefficients, are computed from the buffered speech samples using the well-known normal equations [10]

$$\sum_{j=1}^{p} \phi_{ij} a_j = c_i, \quad 1 \le i \le p \tag{6}$$

where ϕ_{ij} and c_i are defined by

$$\phi_{ij} = \sum_{n=1}^{N} s(n-i)s(n-j), \quad 1 \le i, j \le p,$$

and

$$c_i = \sum_{n=1}^{N} s(n)s(n-i), \quad 1 \le i \le p, \tag{7}$$

respectively and N is the length of the buffer. If the values $s(-p+1) \cdots s(0)$ are assumed zero, the equations are known as the *correlation* equations and the computed values of a_k form a stable filter. If these values are also assumed known from the past, the equations are known as the *covariance* equations. In practice, the coefficients are com-

puted once every 10-20 ms and *frame overlap* and *windowing* are used [10] to ensure that the computed parameters vary smoothly. Although for small frame sizes, the covariance equations give more accurate results, the stability of the resulting filters cannot be guaranteed. A modified version of the covariance solution [11] has been found to give both stable filters and good results with small buffers.

2) Pitch Prediction: In addition to the short-term correlation mentioned above, speech also demonstrates *quasi-periodicity*. For voiced sounds (such as vowels) the short-term spectrum of speech has an envelope largely determined by the short-term correlations and a fine structure arising from the quasi-periodic excitation of the vocal tract. The quasi-periodic nature of voiced speech remains to a large extent in the difference signal obtained after short-term prediction. This periodicity can be removed by further prediction [4]. Let the nth sample of the difference signal after short-term prediction be given by

$$d(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k). \tag{8}$$

The predicted value of $d(n)$ is given by

$$\hat{d}(n) = \sum_{k=-T}^{T} b_k d(n - M + k) \tag{9}$$

where the parameters b_k are the coefficients of a $2T + 1$ tap predictor. The delay M is defined as the delay for which the normalized correlation coefficient between $d(n)$ and $d(n - M)$ is highest. In cases where the signal is periodic, the delay corresponds to a pitch period (or a number of pitch periods). For nonperiodic signals, the delay M becomes random. The coefficients b_k are found by minimizing the mean-squared difference between $d(n)$ and its predicted value by equations similar to (6) [12]. Typically, a 3-tap filter is found sufficient to remove pitch redundancy [4], [11]. It is also possible to construct pitch predictors with noninteger delays [14].

3) Noise Shaping: Traditionally, waveform coders attempt to minimize the mean-squared error (MSE) between the original and predicted waveforms. However, it is well recognized [4] that minimizing the MSE is not equivalent to minimizing the subjective distortion present. In considering perceptual distortion, it is necessary to consider both the short-term spectrum of the noise and its relation to the short-term spectrum of speech. Due to auditory masking, the noise in the high-energy regions of speech (such as formants) is masked by the speech signal itself. Thus the noise components in such regions can have higher energy relative to the noise components in regions where the speech energy is low. Similarly, the auditory system is not equally sensitive to noise in all frequency regions, and higher noise power can be tolerated in the lower sensitivity regions. In most cases these effects can be included in predictive coders by *noise shaping*, i.e., by modifying the short-term spectrum of the noise based on some perceptual criteria.

A number of techniques have been proposed for including noise shaping in predictive coders [4], [11], [13], [15]. Most of these can be shown to be mathematically equivalent since all provide for controlling the noise spectrum in some way, although one form may be preferred over another for implementation reasons or based on ease of parameterization. A block diagram of one such coder [15] is shown in Fig. 5. It consists of a DPCM coder where the quantization noise is fed back within the DPCM loop through a noise-shaping filter whose transfer function determines the spectral shape of the noise. The exact choice of the noise-shaping filter depends on the implementation and the noise masking or other perceptual effects desired; descriptions are available in the literature cited above and elsewhere.

4) Quantization of Side Information: For forward adaptive predictive coders, side information in the form of predictor coefficients and quantizer stepsize is generated, and also has to be quantized prior to transmission; needless to say, the coder and the decoder both use the quantized values of these parameters. The predictor coefficients a_k are almost never quantized directly because they are highly sensitive to quantization and can lead to unstable filters. Transformed values of these coefficients such as *log-area coefficients* or *reflection coefficients* [10] are computed and quantized instead. A simple approach for efficient quantization is to match the individual quantizers to the statistics of the corresponding parameters and distribute the available bits based on the variance of the parameters [16]. It is also possible to exploit dependencies that exist within the coefficient vectors as well as the dependencies between successive vectors by using transform coding or differential coding of coefficient vectors [17]. Coefficients can also be efficiently quantized using *vector quantization* [18], where codebooks containing representative vectors are computed and stored both at the coder and the decoder. The

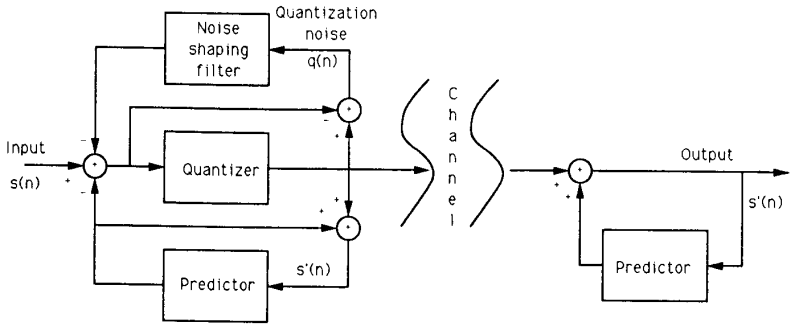


Fig. 5. Block diagram of an APC coder with noise shaping [15]. The coder is similar to the DPCM coder shown in Fig. 3 except that the quantization noise is fed back into the DPCM loop through a noise shaping filter.

coder then simply transmits the index of the appropriate vector to the decoder. Vector quantization for speech coding is reviewed at length in [18].

B. Multipulse and Code-Excited Coders

Predictive coders reduce the bit rate by removing redundancy from the speech signals by linear prediction; and then transmitting the quantized parameters of the predictor as well as the quantized prediction residual. It is, however, very difficult to quantize the prediction residual accurately at rates less than 2–3 bits/sample. Below these rates, the quantization error starts showing significant correlation with the speech signal. As a result the computed predictors no longer remain optimal and the coding gain drops. Both *multipulse* (MPLP) and *code-excited* (CELP) coders get around this problem by ignoring the prediction residual and explicitly modeling the input to the linear predictor using an analysis-by-synthesis loop that minimizes a frequency-weighted mean-squared error between the coder input and the decoder output. This analysis-by-synthesis loop can be seen in Fig. 6, which shows the block diagram of a multipulse coder. The linear predictor (possibly containing a pitch predictor as well as the spectral predictor) produces synthetic speech $\hat{s}(n)$ which is subtracted from the original speech $s(n)$ to form the difference signal $d(n)$. The difference signal is frequency weighted (to achieve noise shaping) to obtain the error signal $e(n)$ which is fed to the excitation generator. The excitation generator forms the heart of both the multipulse coder and the CELP coder and creates an excitation which minimizes the energy in the weighted error signal $e(n)$. For linear noise weighting filters, the computation complexity is reduced somewhat if the weighting filter is moved before the summing junction and both $s(n)$ and $\hat{s}(n)$ are filtered rather than the difference signal $d(n)$ as shown in Fig. 6.

1) Multipulse Coders: In multipulse coders [19], the excitation is modeled as a small number of pulses whose locations and amplitudes are computed to minimize the error energy. The quantized values of these locations and amplitudes form the excitation information to be sent to the receiver along with the filter parameters. Since only a small number of pulses are needed per frame, the bit rate required is lower than for APC coders providing the same quality.

If the input is a series of m pulses of amplitude $\beta_0, \beta_1, \cdots, \beta_{m-1}$ at times $n_0, n_1, \cdots, n_{m-1}$ respectively, the predictor output $\hat{y}(n)$ for the present frame is given by [20]

$$\hat{y}(n) = \sum_{k=0}^{m-1} \beta_k h(n - n_k) + \hat{y}_0(n), \qquad 0 \le n < N, \tag{10}$$

where $\hat{y}_0(n)$ is the contribution to $\hat{y}(n)$ from the filter memory from previous frames and $h(n)$ is the nth sample of the impulse response of the cascade formed by the predictor and the noise weighting filter.

If the pulse locations $n_0, n_1, \cdots, n_{m-1}$ are known, then the pulse amplitudes $\beta_0, \beta_1, \cdots, \beta_{m-1}$ are easily found by equations similar to (6), that is,

$$\sum_{k=0}^{m-1} \beta_k \alpha(n_k, n_j) = c(n_j), \qquad 0 \le j < m \tag{11}$$

where α is the autocorrelation of the impulse response $h(n)$, that is,

$$\alpha(i, j) = \sum_{n=0}^{N-1} h(n - i)\, h(n - j), \qquad 0 \le i, j < N \tag{12}$$

and c is the cross-correlation between the impulse response $h(n)$ and the signal $\bar{y}(n) = y(n) - \hat{y}_0(n)$, that is,

$$c(i) = \sum_{n=0}^{N-1} \bar{y}(n)\, h(n - i), \qquad 0 \le i < N. \tag{13}$$

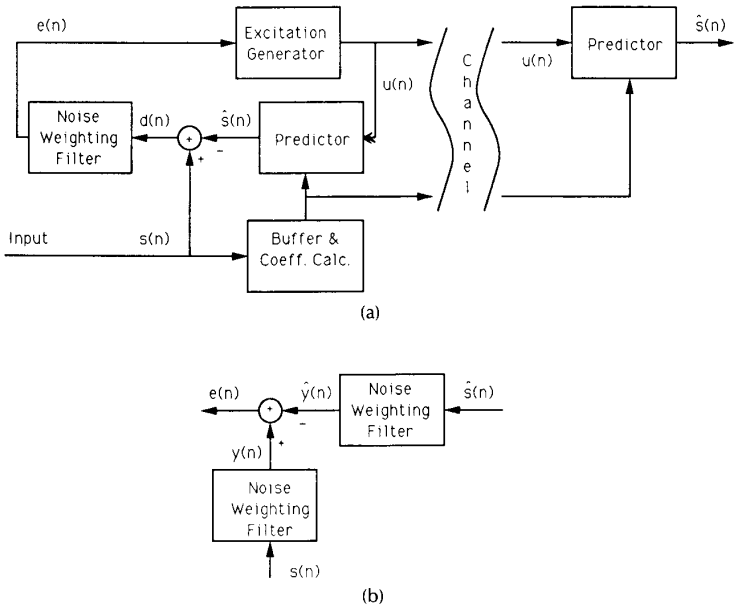


Fig. 6. (a) Block diagram of a multipulse coder. (b) An alternate way of computing the weighted error signal.

However, computation of the pulse locations n_0, n_1, $\cdots$, n_{m-1} is a combinatorial problem and does not have a closed-form solution. We need to choose m of the possible N locations such that the corresponding solution of (11) results in minimum error.

An exhaustive search for pulse locations quickly becomes impractical as the number of pulses m increases. The required computation can be kept within reasonable bounds by searching for pulse locations in stages; at each stage a new pulse is found while previously found pulses are assumed known. This reduces the computation required to m searches of order N [19]. Many variations of this method as well as other methods are available to compute the pulses efficiently [20]–[24]. Multipulse coders can produce very high quality output at 16 kbits/s for telephone bandwidth speech where careful listening is required to tell the original apart from the coded speech. At 9.6 kbits/s, some distortion is perceptible in the output of these coders, although it is not enough to impair normal conversation. Typically, the filter parameters take up about 2–3 kbits/s and the excitation parameters (pulse amplitudes and locations) use the remaining bits. At rates higher than 16 kbits/s, the quality improvement does not justify the additional computation complexity of multipulse coders. Although multipulse coders have been operated at bit rates as low as 4.8 kbits/s [25], quantization of the excitation parameters becomes increasingly difficult and output quality suffers as a consequence.

2) Code-excited Coders: Roughly two-thirds of the bits required in multipulse coders are taken up by the excitation parameters. However, it is known that after both spectral prediction and pitch prediction, the residual signal from speech signals is approximately gaussian distributed [4], [26]. Thus, it is possible to replace the prediction residual by a properly chosen gaussian random sequence. This idea is used in the excitation generator in CELP coders [26] by vector quantizing the excitation (Fig. 7). The excitation generator now consists of a codebook of random excitation sequences known both to the encoder and the decoder. At each step, all excitation sequences in the codebook are filtered through the linear predictor, which consists of both the pitch predictor and the spectral predictor. Note that although multipulse coders can operate without the pitch predictor because the excitation for multipulse coders is relatively unrestricted, CELP coders require the pitch predictor to obtain quasi-periodic outputs. For each waveform in the codebook, the excitation gain is computed to minimize the energy in the weighted error signal. The excitation waveform which results in the smallest error is selected and its index is sent to the receiver along with the excitation gain value and the filter parameters. Because the excitation is now represented only by the gain value and the codebook index, it can be coded very efficiently and high quality can be obtained at rates as low as 4.8 kbits/s.

Due to the closed-loop search in the analysis-by-synthesis technique, the gaussian assumption is not really necessary and other forms of excitation sequences can be employed. For example, multiple codebooks with both pulse-like and noise-like sequences can be used and the excitation values can be limited to have ternary values [27]. It is also possible to structure codebooks in tree or trelis structures [28], [29] which allow efficient searches. A very large fraction of current research [30] in low bit-rate coding is focused on designing CELP coders, constructing "good" excitation codebooks and obtaining efficient procedures to search through these codebooks to enable implementation of these coders on real-time hardware.

C. Frequency Domain Coders

Time domain coders treat the speech signal as a single full-band signal. Predictive coders operate by removing the time redundancy in the signal by using linear prediction. The main difference in the different algorithms lies in the degree of prediction and the quantization strategies. Frequency domain coders, on the other hand, divide the signal into a number of separate frequency components and encode these components separately. The primary advantage of frequency-domain coders lies in the ability to allocate bits dynamically to frequency components where they are needed most. Frequency-domain coders take one of two basic forms: sub-band coders or transform coders.

1) Sub-Band Coders: Figure 8 shows the block diagram of a sub-band coder. Here the speech signal is divided into typically four to eight bands by a set of bandpass filters. The signal in each band is translated to zero frequency and sampled (or resampled) at its Nyquist rate (twice the width of the band), encoded using an ADPCM encoder, multiplexed, and transmitted to the decoder. At the decoder, the signals in the different banks are decoded, modulated back to their original frequency locations and summed to form the decoder output. In digital implementations, the mod-

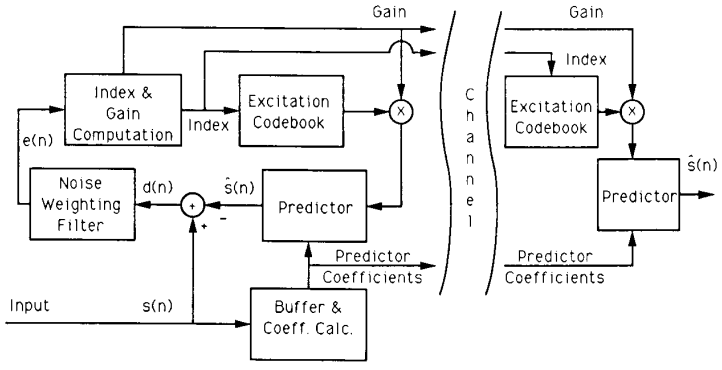


Fig. 7. Block diagram of a CELP coder. The coder is similar to Fig. 6 with the excitation generator replaced by a codebook containing samples from a random process.

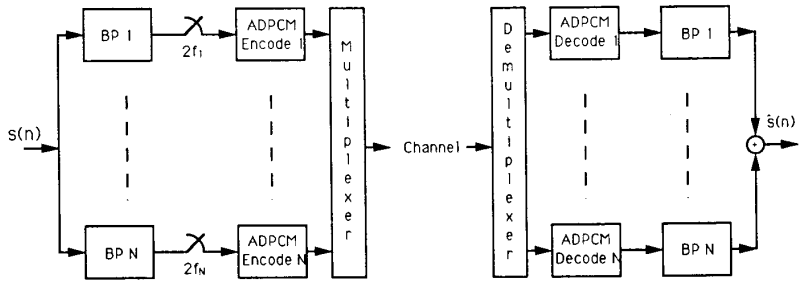


Fig. 8. Block diagram of a sub-band coder. The incoming signal is split into N bands, each band is independently coded and transmit over the channel. The decoder decodes the information received for each band and combines the outputs to form the output signal.

ulation at the decoder is achieved by using bandpass filters at the appropriate frequencies.

In practice, the bandpass filtering and sample-rate conversions are done together by techniques similar to single-side-band modulation or integer-band sampling [31]. The most complicated parts of the sub-band coder are the analysis and reconstruction filter banks. The bands may have equal bandwidth or may have variable bandwidth with the bandwidth increasing with the center frequency of the band. The latter scheme takes into account the response of the auditory system, where the frequency sensitivity falls off with frequency [2]. This choice, however, is not very critical in the presence of dynamic bit allocation [32] which offers the possibility of encoding the signal in various bands with differing fidelity. It is also desirable to avoid aliasing between frequency components located in different bands. Two approaches may be taken to achieve this. First, the filters may be designed with sharp cutoffs to achieve non-overlapping bands. While this reduces the aliasing and offers reduced sampling rates for each band, it also causes inter-band frequency gaps in the output. These gaps are of nonzero width in practical filters and cause a reverberant quality in low bit-rate applications. If the filters have overlapping characteristics, the inter-band gaps can be avoided, but aliasing between the signals in the different bands can lead to distortion in the output speech. These problems may be reduced by using *quadrature mirror filters* (QMF) [33], [34]. These filters have the property that any aliasing introduced at the coder is exactly canceled out at the decoder in the absence of quantization.

Since separate adaptive quantizers are used in different bands, the quantization noise in each band can be controlled independently (subject to the overall bit-rate constraints). Thus bands with low energy can have small quantizer step-sizes and thus lower quantization noise. By dynamically allocating bits among bands [32], [35], the shape of the quantization noise can be controlled based on perceptual criteria. A two-band sub-band coder has been standardized for encoding 7 kHz bandwidth speech at 64 kbits/s by the CCITT [36]. Sub-band coders with noise shaping are also popular [37], [38] for transmission and storage of high-quality audio (15–20 kHz bandwidth) a bit rates of 128–256 kbits/s.

2) Transform Coders: Transform coders [2] window the speech signal, block transform it using an invertible time-to-frequency transform, then quantize and transmit the frequency domain parameters resulting from the transform. At the receiver, the quantized coefficients are inverse-transformed to get the output speech segment. If the quantization process is adaptive, the resulting class of coders is known as *adaptive transform coders* (ATC) [39]. Figure 9 shows this process as a block diagram.

Although many good transforms exist, the *discrete cosine transform* (DCT) is usually used as the time-to-frequency transform in transform coders. It is attractive since it is a good fit to the optimal *Karhunen-Loeve transform* (KLT) in a long-term average sense. It is also easy to compute and has even symmetry which helps reduce end effects at block edges. The DCT and its inverse (IDCT) of an N-point sequence $x(n)$ are defined as

$$X(k) = \sum_{n=0}^{N-1} x(n)\, g(k) \cos\left[\frac{(2n+1)k\pi}{2N}\right], \qquad 0 \le k < N, \tag{14}$$

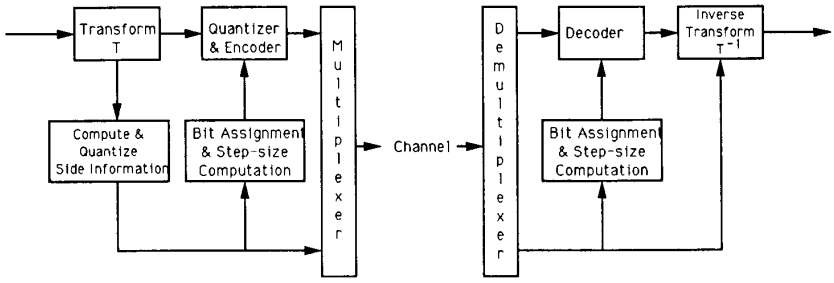


Fig. 9. Block diagram of an adaptive transform coder [1]. The incoming signal is block transformed by a time-to-frequency transform and the resulting transform coefficients are quantized and transmit to the receiver which computes the output signal using the inverse transform.

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, g(k) \cos\left[\frac{(2n+1)k\pi}{2N}\right], \quad 0 \le n < N, \tag{15}$$

where $g(0) = 1$ and $g(k) = \sqrt{2}$, $1 \le k < N$. Once again, as for computing the filter parameters for predictive coders, the input signal is usually analyzed using overlapping analysis windows [39]. The choice of the analysis window is important in controlling boundary effects. The analysis window is usually tapered to zero at the block edges. By allowing a small overlap between successive blocks such that the sum of the overlapped windows is always unity, an additional reduction in the end effect is achieved.

The transform coefficients are usually individually quantized using a b_i bit uniform quantizer for the ith coefficient. The bit assignments are constrainted by

$$B = \sum_{i=0}^{N-1} b_i \tag{16}$$

where B is the number of bits available per block.

If the transform coefficients were stationary gaussian random variables with variances σ_i, $0 \le i < N$, then the optimal bit assignment for the ith coefficient is given by [1]

$$b_i = \delta + 0.5 \log_2 \frac{W_i \sigma_i^2}{D^*} \quad 0 \le i < N \tag{17}$$

where δ is a correction term that reflects the performance of practical quantizers, and W_i is a weighting function allowing noise shaping. D^* is the weighted noise power

$$D^* = \frac{1}{N} \sum_{i=0}^{N-1} W_i E_i^2 \tag{18}$$

and E_i is the noise introduced in the ith coefficient due to quantization. In ATC, the bit assignment is varied from block to block based on the spectral envelope of the signal in the block. Other schemes which use vector quantization [40], [41] are also possible. Like sub-band coders, ATC coders are also currently popular for encoding high-quality audio [42]–[44].

D. Quality Comparisons

The techniques discussed above cover a broad range of transmission rates from 64 kbits/s down to 4.8 kbits/s. Comparison of the quality obtained by different coders operating at these diverse rates and making tradeoffs between quality and bit-rate is a difficult problem. At higher bit rates objective measures such as the SNR can be used. However, subjective tests are usually required to obtain measures of speech quality when comparing speech coders. This is especially true at the lower bit rates where perceptual criteria and noise shaping are used to improve performance and coder parameters are optimized for speech. Careful analysis is required if the coders are used on a channel where nonspeech waveforms may also be present.

A commonly used subjective measure is the so-called *mean opinion score* (MOS) scale [45], which uses a rating scale of 1–5 with 5 being the highest score. A score of 4.0 on this scale is used to denote high quality or near transparent quality and a score of 3.5 denotes communications quality. At scores lower than 3.0 distortion is usually perceptible and the coded speech may in fact sound synthetic.

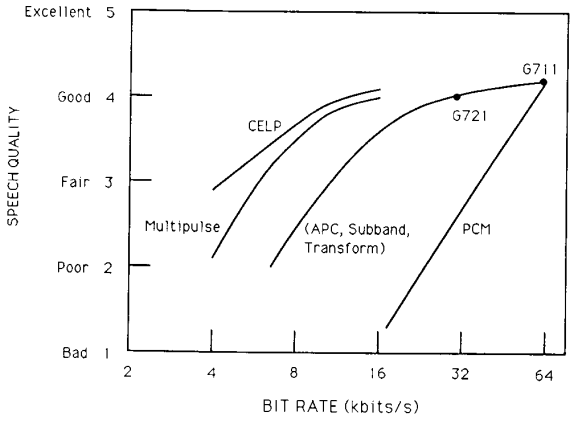


Fig. 10. Comparison of subjective quality for different coders [45].

Figure 10 shows the speech quality as a function of the bit rate for different coders [45]. Since even in a single class of coders, different implementations can have very different quality, the curves in the figure only represent average performance.

E. Summary

Transmission channels employing 32–64 kbits/s rates usually use μ-law or A-law PCM or ADPCM. Coders operating at 4.8–16 kbits/s are primarily CELP and/or multipulse coders with CELP coders gaining popularity in the last few years. At the intermediate rates of 16–32 kbits/s, there are a number of options available including APC, sub-band coders and transform coders although these coders can be used at lower bit rates. For example, vector quantized versions of APC have been used in mobile satellite communications with bit rates as low as 4.8 kbits/s [46], [47].

Standards exist primarily for the higher bit rates (32 and 64 kbits/s) and the CCITT is currently evaluating a 16 kbits/s voice standard for special applications. Variants of multipulse coders and CELP coders have been standardized for use in mobile radio [48] and secure voice applications [49]. For audio applications, sub-band coding has been used for digital audio broadcasting [50] and the International Organization for Standardization (ISO) is considering sub-band and transform coders operating in the 128–256 kbits/s range as standardization candidates for encoding high-quality audio for storage applications. As they mature, these coders now offer a range of options for use in communication and broadcast over channels where the channel bandwidth is limited.

III. Coding of Image and Video Signals

This section is concerned with the digital compression of color images and video signals. With the advent of color television and the introduction of composite signals (NTSC, PAL, and SECAM), analog bandwidth reduction techniques have been in widespread use for video signals [51]. Recently, in order to overcome the fragility and the limitations of the composite signal, mixed analog-digital techniques have been proposed that rely on digital processing of the video signal (bandwidth reduction, adaptive sampling, component multiplexing) and yield analog signals that can be

transmitted over terrestrial and satellite analog channels. The most notable of those analog-digital techniques are the MAC (multiplexed analog component) signals [52], and the MUSE (multiple sub-Nyquist sample encoding) [53] system for high-definition television (HDTV). In MUSE, the image sampling rate is converted from the original 48.6 MHz to 16.2 MHz, an equivalent of Nyquist bandwidth 8.1 MHz, so that a single analog satellite channel of bandwidth 24/27 MHz can accommodate the broadcast of the frequency-modulated image signals. The sample rate conversion involves decimation and interpolation of line and frame offset sampling patterns, that is, a quincunx structure, to reduce the aliasing effect [53].

For many channels, however, digital transmission is becoming an attractive alternative, and we shall review the bit-rate reduction techniques applicable to a wide range of bit rates and applications, from small-screen videoconferencing to HDTV. Initial work on digital compression of color television signals often started with sampling the composite (NTSC) signal at an integer multiple of the color subcarrier frequency to avoid the intermodulation between the sampling and subcarrier frequencies [57]. While digital compression of sampled composite signals is commonly used in many commercial codecs, the constraint of coding a signal designed for analog transmission is quite restrictive and better results can be obtained by directly coding components. While some of the techniques covered in this paper are applicable to the digitized composite signal, we shall focus on digital compression of component video signals.

The analog video signal is already sampled in both the temporal (field/frame) and the vertical (line) direction. When selecting a sampling frequency for a video signal, the field and line frequencies have to be taken into account so that regular, tridimensional sampling patterns occur. The topic of the three-dimensional sampling of video signals is reviewed in [55]. In order to define a universal representation for digital video, the CCIR, in its Recommendation 601, has specified a family of digital video signals that share important common characteristics [54]:

- Orthogonal (2D) sampling patterns
- Fixed sampling structure (each frame has identical sampling pattern)
- Spatial location of chrominance sample to coincide with luminance samples (co-siting).
- Maximum commonality between 525/60 and 625/50 formats.

Video signals digitized according to CCIR Recommendation 601 [56] are matrixed according to the *Y*, Cr, Cb components, where *Y* represents the luminance and Cr and Cb the chrominance components of the video signal. The studio standard of this recommendation is defined as 4:2:2, because the luminance is sampled at 4 times the smallest common frequency between the 525/60 and 625/50 system and the chrominance at half that rate. With the studio standard, the luminance is sampled at 13.5 MHz, yielding 720 active samples per video line; the 360 chrominance samples are co-sited on the luminance samples. All components digitized according to CCIR 601 are linearly quantized and represented with 8 bits per pixel (the components are also calibrated properly to account for the nonlinear relationship between the signal voltage and the resulting light intensity on a CRT screen [58]). The recommended levels for the luminance go from 16 (black) to 235 (white) and certain bit combinations (0 and 255) are reserved for synchronization. Similarly, the number of PCM levels for the chrominance is limited to 225 (centered around 128).

Digital transmission of video signals has numerous advantages: quality, robustness, ease of encryption, bandwidth efficiency and ease of operation and maintenance. The bit-rate reduction techniques for video have matured significantly in the last decade [59], [60] and a worldwide effort of standardization is underway for digital video transmission [61]. We first cover the techniques of redundancy reduction in the spatial domain, then consider the temporal bit-rate reduction techniques using interframe techniques.

A. Intraframe and Still Image Coding

The removal of spatial redundancy is generally viewed as the principal task in the bit-rate reduction process. The techniques used are similar to those used in speech coding: predictive coding, transform coding, sub-band coding, and vector quantization are also used with digital images and video signals. Although image signals are often modeled as locally stationary, the transient part of the signal consisting of "contours" or "edges" plays a unique role in visual perception that has no analog in speech perception. Thus, linear prediction by itself does not result in prediction gains comparable to those obtainable in speech coding. Linear predictors are, however, used for image coding, in particular for relatively high quality, high bit-rate solutions.

1) Predictive Techniques: Predictive techniques, DPCM in particular, have been the first to be used for bit-rate reduction of video signals. The block diagram of the basic DPCM system is similar to the structure for speech shown in Fig. 3. The input signal now becomes a vector consisting of the video component signals and the predictor is replaced by a spatial predictor in one or two dimensions. While two-dimensional predictors fail to perform as well as one could anticipate from a theoretical stationary model, they still outperform one-dimensional predictors. The neighborhood used for predicting the value of the current pixel *x* in simple linear predictors is illustrated in Fig. 11 and

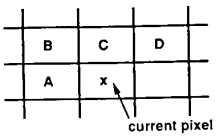


Fig. 11. Pel neighborhood used in intraframe prediction; *x* is the current pixel.

some of the most popular predictors are:

$$\hat{x} = A \tag{19}$$

$$\hat{x} = 1/2\ A + 1/4\ C + 1/4\ D \tag{20}$$

$$\hat{x} = 1/2\ A + 1/8\ B + 1/4\ C + 1/8\ D \tag{21}$$

$$\hat{x} = C \tag{22}$$

where $\hat{x}$ is the predicted value and A, B, C, D are the values of neighboring pixels as shown in the figure. The predictors given in (19) and (22) are the *previous pel predictor* and the *previous line predictor* respectively, while the predictors

(20) and (21) are good fixed predictors with simple "power-of-2" coefficients suitable for a hardware implementation. While many linear combinations of {A, B, C, D} give good predictors, in order to avoid instabilities and limit cycles, Pirsch [62] warned against using negative coefficients when the quantizer in the feedback loop is relatively coarse. To improve the compression of a DPCM system, entropy coding is usually used on the quantized prediction residual. Other ways of improving the basic performance of a DPCM coder are by using subjectively optimized quantizers and adaptive predictors.

1) Subjectively optimized quantizer: Quantizers for DPCM systems have been extensively studied and rely on subjective criteria such as visibility threshold or masking function [63]. In a DPCM video coder the impairments introduced by quantization are *granular noise*, *edge busyness*, and *slope overload*. The masking functions designed by careful experiments [63] or by means of models of human vision represent the envelope of the just-not-visible (threshold) quantization error. Note that the quantization error is invisible as long as it remains below the masking curve (see Fig. 12).

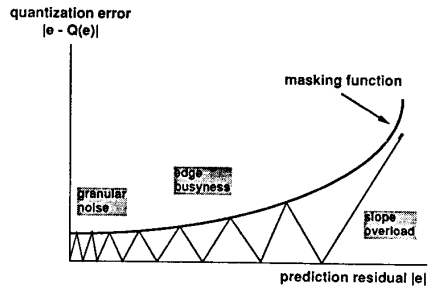


Fig. 12. Masking function for subjectively optimized quantizers is the envelope of the just-not-visible quantization error curve. In a DPCM video coder, there are three types of impairments: granular noise, edge busyness, and slope overload. Quantization error is invisible as long as it remains below the masking curve.

Since subthreshold coding of the prediction error requires fairly fine quantizers (and consequently many bits), the quantization error is frequently above threshold in practice, and noise shaping, similar to the one in Fig. 5 for speech coding, can be used to make the quantization error pattern less visible. Noise shaping is similar to the *error diffusion technique* used in digital halftoning [66], but the quantization error is fed back within a DPCM loop. It has been shown [67] that noise shaping shifts the energy of the quantization error spectrum toward the higher end of the spectrum where it is less visible. There are also drawbacks to noise shaping, in particular, the entropy of the quantization error is increased; on the other hand, it proves to be an effective technique when using quantizers with a limited number of levels (5 bits) [67]. Because this technique is relatively simple to implement at high speed, an intraframe HDTV coder using subsampling and noise-shaping DPCM has been demonstrated; it operates in the range of 120–140 Mbit/s [68].

2) Adaptive and non-linear predictors: Another approach to increasing the performance of the DPCM coder is to refine the predictor. In [64], an anisotropic linear predictor is proposed that can take into account possible edges in the image. Another way of improving the prediction is to use multiple predictors and switch to the one that is likely to be the best based on the context. Because of the structure of the DPCM loop, if the context is derived from the past quantized prediction error, no additional information needs to be transmitted to the decoder. A particular implementation of this class of predictors relies on medians: the "best predictor" is the median of the outputs of all candidate predictors [65].

Intra-frame predictors have played an important role in image coding [57], [59], [60] and are very simple to implement. However, their compression capability is limited and increasingly, they are not used alone, but combined with interframe predictors [65], [67] or with transform coding [64]. With the studio standard 4:2:2 video signal, intraframe DPCM can compress the signal in the range of 90–140 Mbits/s. Recent trends in standardization, however, have favored lower bit rates such as 34 and 45 Mbits/s, which cannot be achieved easily with intraframe DPCM [61], [72].

2) Transform Coding Techniques: In *transform coding*, the image (field, frame) is partitioned into nonoverlapping blocks, which may be as small as 4×4 or as large as 32×32 pixels. Each block is then transformed by an orthogonal transform into a spectral-like representation. The orthogonal transform (DFT, DCT, Hadamard, Karhunen-Loeve) is chosen for its ability to concentrate the energy of the signal in the "low frequencies" in the transform domain. If the criteria for goodness of the transform is the mean square error (MSE), then the Karhunen-Loeve transform is the optimum; it is, however, not easily implementable. If one assumes a stationary model for the image, and furthermore that it is "first-order Markov" with a correlation parameter p larger than 0.6, then the DCT is very close to the "optimal" Karhunen-Loeve transform, and this holds true even for small block sizes commonly used for image coding. Fast algorithms are available for computing the DCT, and VLSI implementations have been demonstrated at video rates [75].

The theoretical justification for the DCT has some flaws, however. Although it is true that when an image is partitioned in small blocks (say 8×8) the image in many blocks is a smooth "high-correlation-like surface," the critical blocks containing most of the image information (edge, characteristic textures) tend to fail the theoretical model. Nevertheless, the basis functions of the DCT are essentially well-behaved (symmetric and all derived by modulation of a rectangular window), and the DCT tends to outperform other transforms on edges as well. Other orthogonal transforms have been suggested in the literature [70], but with improved techniques for quantizing DCT coefficients [71] and technological advances in VLSI [75], the DCT has established itself as the transform of choice. The choice of the block size is the result of a trade-off between the compression efficiency and the local stationarity of the signal. In current systems [74], [97] the 8×8 block size is usually preferred; this block size is particularly appropriate when using combined run-length and amplitude entropy coding [74].

Increasing experience with transform coding, in particular DCT coding, has shown that there was much to gain by using subjectively adaptive quantization [71]. If one considers a given signal (for example, CCIR 601), a particular class of displays, fixed illumination conditions, and a view-

ing distance (for example, 4 times the screen height), it is possible to determine subjectively the visibility threshold for each individual DCT coefficient. When used carefully, visual weighting techniques that use coarser quantizers for the higher frequency coefficients greatly improve the performance of the DCT [74]. It is also possible to exploit visual masking phenomena, by noticing that for blocks where the energy is spread among many coefficients (i.e., noise-type textures) the quantization errors are largely masked [73]; on such blocks coarser quantizers are appropriate. For video coding applications with the DCT, it should be kept in mind that the temporal visibility of the quantization noise is different from the spatial visibility within a single frame/field and small quantization errors can give rise to a "busyness" sometimes qualified as "mosquito effects." Thus, temporal effects should be taken in consideration when using subjectively optimized quantization.

The DCT is considerably more efficient in compressing an image "intraframe" than are DPCM techniques. Visually satisfying results for video compression can be obtained for as low as 2 bits/pel while equivalent DPCM results typically require 3 to 4 bits/pel for the *Y*, Cr and Cb components. Results with higher compression and barely visible coding artifacts can be obtained for still images in the range of 0.75 to 1 bit/pel [74]. A proposal has been made [76] for the signal transmission in both pseudo-components and 4:2:2 format with intra-frame DCT in the range of 30 to 45 Mbit/s. Many coders operating at a lower bit rate [97] or with a higher quality [72] are based on DCT but combine it with interframe prediction.

3) Interpolative Techniques: Even with two-dimensional predictors, DPCM systems have inherent limitations for coding two-dimensional signals. Because DPCM predictors are causal, they rely heavily on the raster scan order. An alternative to causal predictors is to use "non-causal" predictors, which amounts to using interpolators as predictors. In this case, a hierarchy of increasing resolution grids is obtained: the interpolation error between the image at resolution k, I_k and the image at resolution $k+1$, I_{k+1} is the signal to be coded. Interpolative techniques give rise to hierarchical structures, also called *pyramids* [77]. Because the quantized image at one resolution can be used (after interpolation) as a predictor for the next higher resolution, a coding loop analogous to the DPCM loop can be introduced (Fig. 13), but in this case, it is the resolution index

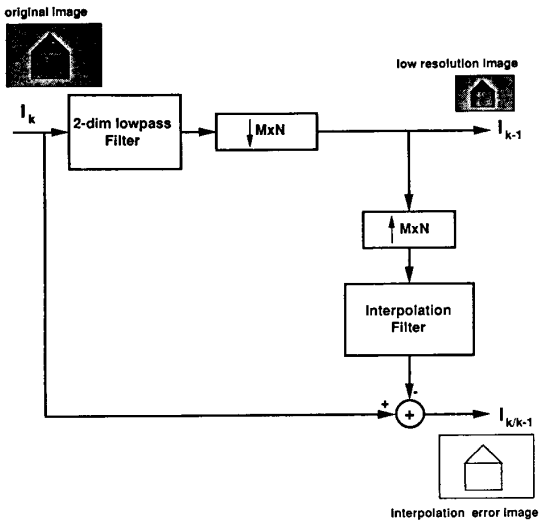


Fig. 13. A two-stage pyramid. The total number of pixels in both lower-resolution and interpolation error images is higher than the input image.

that is incremented at each stage. A three-stage pyramid structure is shown in Fig. 14. The pyramidal structures give relatively efficient codes, and compression comparable to that of transform coders can be obtained. However, the technical difficulty of managing multiple signals with different resolutions in a real-time video system is such that few video coders have been realized based on pyramids.

Another coding concept involving multiple resolutions is two-dimensional sub-band coding [78], [79]. Sub-band coders have been used in speech coding for many years and the concept of filter banks is closely related to the ideas of interpolation and filtering. As in speech, the codec contains two phases: analysis and synthesis. During the analysis part, the incoming video signal is passed through N linear filters, yielding N-channels. Each channel is then decimated by a factor M; when the decimation factor is equal to the number of channels ($M = N$), the filter bank is said to be critically decimated, and the number of samples remains constant throughout the system. The N filtered decimated signals are then coded independently with DPCM coding [79],

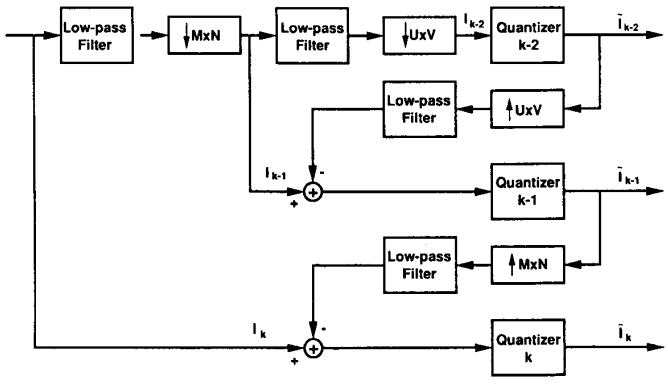


Fig. 14. A three-stage recursive structure with decimation/interpolation and quantization.

DPCM-PCM [80], or transform coding and PCM [82]. At the decoder each channel is decoded, interpolated and the channels are added together (*synthesis*) to recreate the original signal. The theory of filter banks [78], [84] determines the conditions under which the original signal can be reconstructed perfectly or an alias-free approximation can be obtained. For two-channel filter banks (yielding four channels with separable two-dimensional filters), quadrature mirror filters (QMF) have been popular [79], [80]. A comparison of many filter banks, taking into account compression efficiency and complexity, is available in [85].

When the number of channels is small (for example, the separable four-channel case, with low-low, low-high, high-low, and high-high channels (see Fig. 15)), sub-band coding is closely related to a two-stage pyramid, with the difference that fewer samples need to be coded in the (critically decimated) sub-band case, and the "high-resolution" interpolation error of the pyramid is split into directional subchannels (vertical, horizontal, and diagonal), thus contributing more to decorrelation in the signal. An important factor in sub-band techniques applied to video signals is the relatively low complexity of the filters needed to split the signal into useful sub-bands [81]. When using perfect reconstruction techniques, one can design low complexity filters with "power-of-two" coefficients that are suitable for very high-speed implementations. For HDTV coding applications, sub-band techniques also present the advantage that much of the processing is performed at much lower clock rates than the clock of the incoming video signal. An intra-field combination of sub-band coding and transform coding has been proposed in [82] to compress an HDTV signal down to 120 Mbits/s.

When the number of channels is large (such as a separable 8 × 8 filter bank), sub-band coding becomes much closer to transform coding. Orthogonal transforms can be viewed as a particular, degenerate case of perfect reconstruction filter banks [84], and some recently published work [83] has shown that generalized transforms can have performance comparable to the DCT with a possible reduction of the so-called blocking effects. Generalized transforms derived from filter banks, however, do not have the maturity of transform coding and more work is needed, in particular in the field of visual perception, before one can draw conclusions about their importance.

4) Block Coding and Vector Quantization: Transform coding techniques are a particular case of block coding techniques that rely upon a particular "frequency" domain interpretation. Spatial techniques based on partitioning the image in blocks can be equally efficient. A very simple block coding scheme is the *adaptive dynamic range coding* (ADRC) technique [86]. In ADRC, the minimum and maximum values of the block are extracted (alternatively, the mean and the range), a quantizer is defined based on the mean and the range, and each picture element in the block is quantized with this quantizer. Given a simple block quantization scheme, two directions can be chosen to improve the compression: make the block size adaptive and use additional coding to further exploit the redundancy.

1) Adaptive block size block coding scheme: Making the block size adaptive is a good way to code the near uniform regions in a picture with a few bits and to concentrate more bits on those regions that contain edges and difficult textures. The *quadtree* data structure is a representation that can be used for adaptive block coding [90]. In a quadtree the parent of four nodes is associated with their average: One can view the quadtree as a multiple resolution decomposition much like recursive pyramids (Fig. 14), but where the depth of the decomposition is controlled by the image content. However, the filters are very short, and aliasing is ever present.

2) Vector quantization: Vector quantization (VQ) can also be used to effectively reduce the data redundancy for image coding. In the simplest scheme, neighboring pixels are jointly treated as a vector and mapped, according to the minimum distance criterion, into one of the finite codevectors stored as a codebook, and the corresponding codevector index in the codebook, which representsd the coded information, is sent to the decoder. A good set of training vectors is needed to generate a "robust" codebook for a wide variety of images; several popular methods can be found in [92]. However, the pixel values are seldom used directly in practical schemes; the residuals obtained from some form of prediction are preferred instead. In [87], the vectors are obtained from a 4 × 4 block of samples by subtracting the block average. In order to achieve a good visual quality with this simple scheme, the codebooks have to be quite large. The drawbacks of a large codebook are clear: heavy computing load if an exhaustive search is used and heavy memory requirement to store the codebook. In order to remedy these drawbacks, much research has been directed toward structuring the codebooks and/or obtaining easy-to-code residual signals [88], [91]. Combining adap-

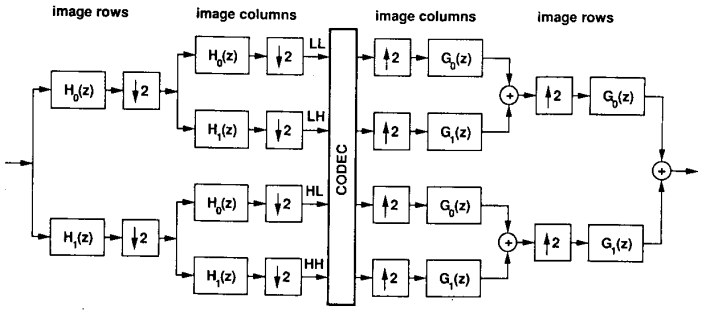


Fig. 15. A four-band analysis and synthesis of digital images. $H_i(z)$ and $G_i(z)$ represent the analysis and synthesis filters, respectively. $H_0(z)$ is a low-pass filter, while $H_1(z)$ a high-pass filter. The four sub-band—low-low (LL), high-low (HL), low-high (LH), and high-high (HH)—are obtained by applying $H_i(z)$ with respect to image rows and columns.

tive block decomposition (quadtree) with VQ leads to excellent results [91]; however, there is a cost in complexity, and many adaptive VQ techniques have been limited to still images. For low bit-rate video coding (from 56 kbit/s to 256 kbits/s), a form of spatially adaptive hierarchical VQ has been proposed and implemented on a programmable processor [89].

B. Interframe Coding

After the removal of intraframe redundancy, many video coding algorithms take advantage of the interframe correlation. The correlation between frames is extremely high in teleconferencing applications, where "head and shoulder" images and fixed backgrounds dominate. The simplest way to exploit a fixed background is to update only the moving parts of the picture. Conditional replenishment techniques, however, are only moderately successful at exploiting the interframe correlation, since they fail to be effective in the presence of a slow global motion.

Interframe techniques have been first used for teleconferencing and with reduced frame rate noninterlaced formats. For interlaced formats such as CCIR 525/60 and 625/50, the "interframe" prediction can choose between a previous frame predictor (with the identical sampling structure as the current field), a previous field predictor (closer in time), and the intrafield predictor (when temporal prediction fails). The best predictor depends on the amount of motion:

- Interframe prediction: no motion
- Interfield prediction: slow to moderate motion
- Intraframe prediction: heavy motion

In DPCM-based interframe algorithms, the median of the three predictors can be used [65]. With hybrid algorithms where the prediction error is transform coded, the decision can be transmitted as an overhead since there is only one decision per block, and the predictor can be chosen by measuring the "energy" of the prediction errors.

The use of interframe prediction is a natural way to exploit the temporal correlation of video signal and increasingly, interframe prediction implies motion compensation. Computationally efficient motion estimation techniques are the key to interframe prediction; we now review the most common of those techniques.

1) Displacement Estimation

1) Techniques based on spatio-temporal gradients: Spatio-temporal gradient techniques are all based on a model where both the displacement field $\underline{d}(\underline{x})$ and the image field $I_t(\underline{x})$ at location $\underline{x}$ and time (frame) t are continuous functions. In this case, the current frames $I_t(\underline{x})$ and the previous frame $I_{t-1}(\underline{x})$ are matched so as to minimize the square of the *displaced frame difference* (DFD),

$$DFD(\underline{x}, \underline{d}) = I_t(\underline{x}) - I_{t-1}(\underline{x}, \underline{d}). \tag{23}$$

The displacement estimate at location $\underline{x}$ and time t is the value $\underline{d}^*(\underline{x})$ that minimizes the square displaced frame difference $DFD^2(\underline{x}, \underline{d})$. In this case, conventional optimization algorithms are applicable and one can find the optimal displacement recursively (Fig. 16). A general formula for the estimate can be written as:

$$\underline{d}_{k+1} = \underline{d}_k + \kappa \cdot \nabla_{\underline{d}} DFD(\underline{x}, \underline{d}_k)^2 \tag{24}$$

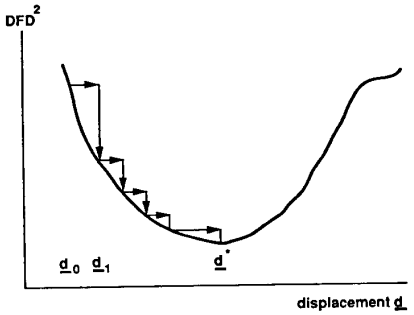


Fig. 16. Pel-recursive motion estimation; find the displacement $\underline{d}^*$ recursively such that DFD^2 is minimized.

where κ is a gain factor that affects the convergence of the estimate. In [93], the gain was taken to be constant, while in [94] and the subsequent literature [60], κ was designed to achieve a faster convergence. Pel-recursive algorithms can use more than one pel in the estimate in order to achieve more robustness to noise [93]. As long as the neighborhood used in the estimation is also known to the decoder, pel-recursive estimators can be integrated in a DPCM loop without the need to send the motion information explicitly.

2) Techniques based on block matching: While techniques based on spatio-temporal gradients provide good estimates of the displacement field, techniques that assume that the displacement field is constant blockwise (uniform translation model) yield estimation techniques with less complexity. Block matching is also particularly well-suited when used in conjunction with a block coding scheme since the motion vector is transmitted together with the coded block. In block-matching motion estimation techniques, the mean square (alternatively, the mean absolute) displaced frame difference is minimized over all possible displacements in a limited search range:

$$\underline{d}^* = \min_{\underline{d} \in R}{}^{-1} \sum_{\underline{x} \in B} DFD(\underline{x}, \underline{d})^2 \tag{25}$$

where B is the block to be matched and R the search range defined as:

$$R = \{\underline{d}: d_x \in [-N, +N], d_y \in [-M, +M]\} \tag{26}$$

This is illustrated in Fig. 17.

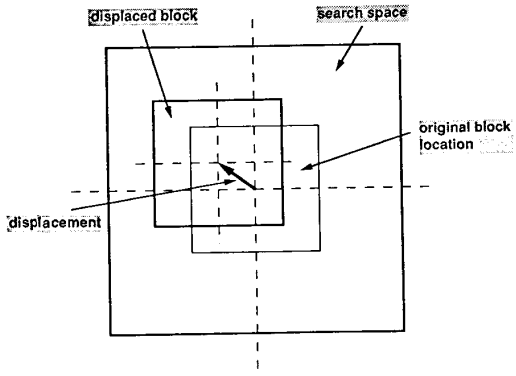


Fig. 17. Motion estimation by block matching. The current block finds the best match from the search space in the previous frame. The motion displacement (vector) is sent to decoder as an overhead.

With block-matching algorithms, one distinguishes approximate searches from exhaustive ones. Initial work with block-matching techniques proposed limited step-by-step suboptimal searches [95], [96]. Recent advances in VLSI technology have made full-search block-matching feasible. The number of matching operations for a given block is potentially very large, but the computational structure is extremely regular and naturally suitable to VLSI [99].

A simple block-matching algorithm such as Eq. (25) is limited to integer accuracy. A slight modification, however, can allow fractional pel accuracy. Fractional search locations are used and the displaced frame difference is evaluated at fractional displacement location by way of linear interpolation. Because the number of opertions for subpixel accuracy block matching is fairly high, hierarchical techniques are used, where the estimate is refined from one resolution (for example, integer) to the next (half pixels). Another advantage of hierarchical block-matching alforithms, besides subpixel accuracy, is the possibility of enforcing some kind of consistency of the motion vector field between blocks that share a common parent at a lower level of the hierarchy. This is of importance when the motion estimate is used for interpolation [98].

2) Motion-Compensated Predictive Coding: Motion-compensated prediction allows a much better used of the strong temporal correlation of video signals. Although initially motion-compensation techniques were motivated by the need of low bit-rate coding for teleconferencing applications [93], [97], motion compensation is being increasingly considered for higher bit-rate and broadcast applications [72].

For coding video at low bit rates, a motion-compensated hybrid coder has been studied extensively by the CCITT Study Group XV [97]. The basic principle is *motion-compensated interframe prediction* and *discrete cosine transform* coding of the prediction error when this error is above a certain threshold. At the beginning of operation, or when the quality of the prediction is very poor, intraframe discrete cosine transform is used. The general structure of a motion-compensated hybrid coder is shown in Fig. 18. The corresponding decoder is shown in Fig. 19.

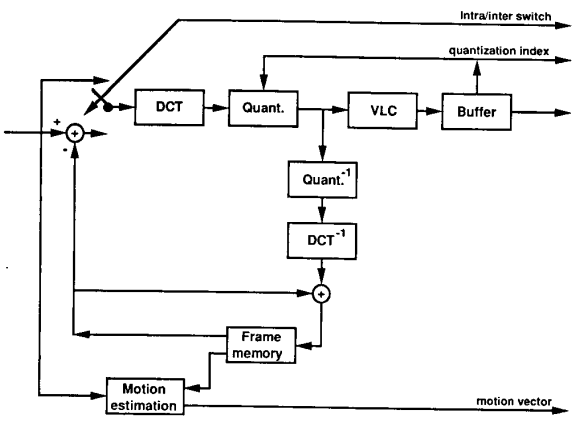


Fig. 18. Motion-compensated hybrid coder; the motion-compensated block difference is coded with DCT schemes if the block difference is above certain threshold; otherwise, the block is intraframe DCT-coded.

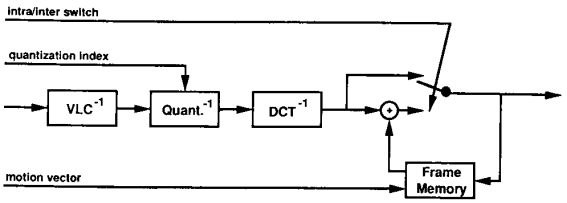


Fig. 19. Decoder corresponding to the motion-compensated hybrid coder of Fig. 18.

Two important parameters in a hybrid coder are the block size (if block-matching algorithms are used) and the accuracy of the motion vector field. The larger the block, the lower the overhead of the motion information, but the more likely it is that the *uniform translation* assumption will fail. Subpixel accuracy is important for accurate prediction of edges, but is computationally more complex, and the prediction gain can be offset by the additional information in the motion vector. For each operating point (frame rate, spatial resolution, target bit-rate), there is an optimal trade-off that can involve computational complexity, block size, and subpixel accuracy.

Although motion-compensated hybrid schemes are usually considered for low transmission rates, the structure of the motion-compensated hybrid coder is efficient at any rate. The most apparent trade-offs is between technology, block size, and accuracy of the motion vector. Often at high bit rate a large part of the coding gain from motion compensation comes from the estimation of a global motion vector (pans, zooms). Recent trends are toward using motion-compensated predictive coding with DCT encoding of the prediction error for contribution quality applications at 34 and 45 Mbits/s [72]. Future progress in integrated circuit technology will allow the use of motion-compensated techniques for HDTV.

3) Motion-Compensated Interpolation: As was mentioned in Section III-A-3, interpolation is a good alternative to prediction, and this is also true of motion-compensated interpolation. A well-conditioned motion vector field obtained by a modified block-matching algorithm [98] allows very efficient motion-compensated interpolation. The technique appears very promising for moderate to high compression from 1 to 10 Mbits/s and for special processing such as frame rate conversion or de-interlacing.

C. Summary

Digital video compression is maturing rapidly, and progress in VLSI has followed new algorithmic developments. A basic structure using motion-compensated hybrid predictive/transform coding can apparently cover the whole range of applications from low bit-rate (64 kbit/s2) videotelephone to advanced television (140-Mbits/s HDTV coding). Behind the common algorithmic tools (and the associated VLSI's), there remain a diversity of formats: various frame rates, raster sizes, interlaced and noninterlaced formats. Techniques that are just emerging, such as multirate signal representations (sub-band and pyramids) and motion-compensated interpolation, may have the potential to unify all video compression techniques within a framework of hierarchical coding algorithms.

Conclusions

In this paper, we reviewed some digital source coding techniques in use for speech, image, and video signals. High-quality speech coders are now available using bit rates as low as 7 kbits/s and research is focused toward implementing good-quality coders at 4.8 kbits/s and below. At these bit rates, most of the trust is toward code-excited LPC and its variants. With the exception of intraframe coding techniques using motion estimation, both image and video coding use the same techniques, with combinations of DCT and sub-band coding yielding among the best results. Motion compensation is invariably used to improve quality of video at the lower bit rates. Still images can be coded at 0.5–1.5 bits/pixel depending on the image content and the quality desired. Bit rates for video vary from 64 kbits/s for videotelephone applications to 150 Mbits/s for HDTV.

References

[1] J. L. Flanagan *et al.*, "Speech coding," *IEEE Trans. Commun.*, vol. COM-27, no. 4, pp. 710–737, 1979.

[2] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice Hall, 1984.

[3] N. S. Jayant, "Digital coding of speech waveforms: PCM, DPCM, and DM quantizers," *Proc. IEEE*, vol. 62, no. 5, May 1974.

[4] B. S. Atal, "Predictive coding of speech at low bit rates,' *IEEE Trans. Commun.*, vol. COM-30, no. 4, pp. 600–614, 1982.

[5] CCITT, Recommendation G.721, "32 kbits/sec adaptive differential pulse code modulation," *Blue Book*, vol. III, fascicle III.3, Oct. 1988.

[6] A. Fukasawa and K. Hosoda, "An advanced 32 kbits/sec ADPCM coding to transmit speech and high speed voiceband data," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, pp. 821–824, 1986.

[7] V. Ramamoorthy, N. S. Jayant, R. V. Cox, and M. M. Sondhi, "Enhancement of ADPCM speech coding with backward adaptive algorithms for postfiltering and noise feedback," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 2, pp. 364–82, 1988.

[8] A. P. Sage and J. J. Melsa, *Estimation Theory with Applications to Estimation and Control*. New York: McGraw-Hill, 1971.

[9] M. L. Honig and D. G. Messerschmitt, *Adaptive Filters: Structures, Algorithms and Applications*. Boston, MA: Kluwer Academic Publishers, 1984.

[10] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*. New York: Springer-Verlag, 1976.

[11] B. S. Atal and M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-27, pp. 247–254, 1979.

[12] R. P. Ramachandran and P. Kabal, "Pitch prediction filters in speech coding," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-37, no. 4, pp. 467–478, 1989.

[13] M. R. Schroeder and B. S. Atal, "Optimizing digital speech coders by exploiting masking properties of the human ear," *J. Acoust. Soc. Am.*, vol. 66, no. 6, pp. 1647–1652, 1979.

[14] P. Kroon and B. S. Atal, "On improving the performance of pitch predictors in speech coding systems," *IEEE Workshop on Speech Coding for Telecommunications*, Vancouver, BC, Canada, Sept. 1989.

[15] J. Makhoul and M. Berouti, "Adaptive noise spectral shaping and entropy coding in predictive coding of speech," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-27, no. 1, pp. 63–73, 1979.

[16] R. Vishwanathan and J. Makhoul, "Quantization properties of transmission parameters in linear predictive systems," *IEEE Trans. Acoust., Speech, Signal Proc.*, pp. 434–446, 1975.

[17] M. R. Sambur, "An efficient linear prediction vocoder," *Bell System Technical Journal*, vol. 54, no. 10, pp. 1693–1723, 1975.

[18] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," *Proc. IEEE*, vol. 73, no. 11, pp. 1551–1588, 1985.

[19] B. S. Atal and J. R. Remde, "A new model of LPC excitation for producing natural-sounding speech at low bit rates," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Paris, France, pp. 614–617, 1980.

[20] S. Singhal and B. S. Atal, "Amplitude optimization and pitch prediction in multipulse coders," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-37, no. 3, pp. 317–327, 1989.

[21] S. Singhal and B. S. Atal, "Improving performance of multipulse coders at low bit rates," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, San Diego, CA, paper 1.3, 1984.

[22] M. Berouti, H. Garten, P. Kabal, and P. Mermelstein, "Efficient computation and encoding of the multipulse excitation for LPC," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, San Diego, CA, paper 10.1, 1984.

[23] J. Lefevre and O. Passien, "Efficient algorithms for obtaining multipulse excitation for LPC coders," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Tampa, FL, pp. 957–960, 1985.

[24] P. Kroon, E. F. Deprettere, and R. J. Sluyter, "Regular-pulse excitation—A novel approach to effective and efficient multipulse coding of speech," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-34, no. 5, pp. 1054–1063, 1986.

[25] K. Ozawa and T. Araseki, "Low bit rate multipulse speech coder with natural speech quality," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, pp. 457–460, 1986.

[26] M. R. Schroeder and B. S. Atal, "Code excited linear prediction (CELP): High-quality speech at very low bit rates," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Tampa, FL, pp. 937–940, 1985.

[27] D. Lin, "New approaches to stochastic coding of speech sources at very low bit rates," *Proc. EUSIPCO*, pp. 445–448, 1986.

[28] M. W. Marcellin, T. R. Fischer, and J. D. Gibson, "Predictive trellis coded quantization of speech," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, New York, NY, pp. 247–250, 1988.

[29] P. Kabal, J.-L. Moncet, and C. C. Chu, "Synthesis filter optimization and coding: Applications to CELP," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, New York, NY, pp. 147–150, 1988.

[30] See for example *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, 1989.

[31] J. M. Tribolet and R. E. Crochiere, "Frequency domain coding of speech," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-27, no. 5, pp. 512–530, 1979.

[32] T. A. Ramstad, "Considerations on quantization and dynamic bit allocation in subband coders," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Tokyo, Japan, pp. 841–844, 1986.

[33] T. P. Barnwell, "Subband coder design incorporating recursive quadrature mirror filters and optimum ADPCM coders," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-30, no. 5, pp. 751–765, 1982.

[34] M. J. T. Smith and T. P. Barnwell, "Exact reconstruction techniques for tree structured subband coders," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-34, no. 3, pp. 434–441, 1986.

[35] Y. Yong and A. Gersho, "Subband vector excitation coding with adaptive bit allocation," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, pp. 743–746, 1989.

[36] CCITT, Recommendation G.722, "7 kHz audio coding within 64 kb/s," *Blue Book*, Vol. III, fascicle III.3, Oct. 1988.

[37] G. Stoll, M. Link, and G. Theile, "Masking-pattern adapted subband coding: Use of the dynamic bit-rate margin," *AES 84th Convention*, Paris, France, 1988, Preprint 2585 (D-5).

[38] S. F. M. Smyth and P. Challener, "An efficient coding scheme for the transmission of high quality music signals," *Br. Telecom. Technol. J.*, vol. 6, no. 2, pp. 60–70, 1988.

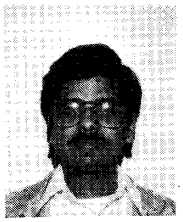
[39] R. Zelinski and P. Noll, "Approaches to adaptive transform speech coding at low bit rates," *IEEE Trans. Acoust., Speech, Signal Proc.*, vol. ASSP-27, pp. 89–95, 1979.

[40] Y. Shoham and A. Gersho, "Pitch synchronous transform coding of speech at 9.6 Kb/s based on vector quantization," *Proc. Intl. Conf. on Communications*, Amsterdam, Netherlands, pp. 1179–1182, 1984.

[41] T. Moriya and M. Honda, "Transform coding of speech with weighted vector quantization," *Proc. Intl. Conf. on Acous-*

tics, Speech, and Signal Processing, Dallas, TX, pp. 1629–1632, 1987.
[42] E. F. Schroeder, H.-J. Platte, and D. Krahe, "MSC': Stereo audio coding with CD-quality at 256 kbits/sec," *IEEE Trans. Consumer Electronics*, vol. CE-33, no. 4, pp. 512–519, 1987.
[43] K. Brandenburg and D. Seitzer, "OCF: Coding high quality audio with data rates of 64 kbit/sec," *AES 85th Convention*, Los Angeles, CA, 1988, Preprint 2723 (H-6).
[44] Y. Mahieux, J. P. Petit, and A. Charbonnier, "Transform coding of audio signals using correlation between successive transform blocks," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, pp. 2021–2024, 1989.
[45] N. S. Jayant, "High quality coding of telephone speech and wideband audio," *IEEE Communications Magazine*, vol. 28, no. 1, pp. 10–20, Jan. 1990.
[46] J. H. Chen, G. Davidson, A. Gersho, and K. Zeger, "Speech coding for the mobile satellite experiment," *Proc. IEEE Int. Conf. on Communications*, Seattle, WA, pp. 756–763, 1987.
[47] G. Davidson, M. Yong, and A. Gersho, "Real time vector excitation coding of speech at 4800 bps," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Dallas, TX, pp. 2189–2192, 1987.
[48] P. Vary, C. Hellwig, and R. Hofmann, "Speech codec for the European mobile radio system," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, New York, NY, pp. 227–230, 1988.
[49] J. P. Campbell, V. C. Welch, and T. Tremain, "An expandable error protected 4800 Kbs CELP coder," *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, Glasgow, Scotland, pp. 735–738, 1989.
[50] G. T. Waters, ed., *Advanced Digital Techniques for UHF Satellite Sound Broadcasting*. Brussels, Belgium: European Broadcasting Union, Aug. 1988.
[51] *Television Engineering Handbook*, K. Blair Benson, Ed. New York: McGraw-Hill, 1985.
[52] M. D. Windram *et al.*, "MAC—A television system for high quality satellite broadcasting," *IBA Experimental And Development Rep.* 118/82, Aug. 1982.
[53] Y. Ninomiya, Y. Ohtsuka, Y. Izumi, S. Gohshi, and Y. Iwadate, "An HDTV broadcasting system utilizing a bandwidth compression technique—MUSE," *IEEE Trans. Broadcasting*, vol. 33, pp. 130–160, Dec. 1987.
[54] F. Kretz and D. Nasse, "Digital television: Transmission and coding," *Proc. IEEE*, vol. 73, Apr. 1985.
[55] E. Dubois, "The sampling and reconstruction of time varying imagery with application in video systems," *Proc. IEEE*, vol. 73, Apr. 1985.
[56] CCIR Recommendation 601, "Encoding parameters of digital television for studies," in *CCIR Recommendation and Reports*, vol. XI, ITU, Geneva, Switzerland, 1982.
[57] J. O. Limb, C. B. Rubinstein, and J. E. Thompson, "Digital coding of color video signals—A review," *IEEE Trans. Commun.*, vol. COM-25, Nov. 1977.
[58] A. N. Netravali and B. G. Haskell, *Digital Pictures*. New York: Plenum Press, 1988.
[59] A. N. Netravali and J. Limb, "Picture coding: A review," *Proc. IEEE*, vol. 68, Mar. 1980.
[60] H. G. Mussmann, P. Pirsch, and H. Grallert, "Advances in picture coding," *Proc. IEEE*, vol. 73, Apr. 1985.
[61] H. Gaggioni and D. Le Gall, "Digital video transmission and coding in the broadband ISDN," *IEEE Trans. Consumer Electronics*, Feb. 1988.
[62] P. Pirsch, "A new Predictor design for DPCM coding of TV signals," *Proc. ICC '80*, June 1980.
[63] ——, "Design of DPCM quantizers for video signals using subjective tests," *IEEE Trans. Commun.*, vol. COM-29, July 1981.
[64] R. Wilson, H. E. Knutsson, and G. H. Granlund, "Anisotropic, nonstationary image estimation and its applications: Part II—Predictive image coding," *IEEE Trans. Commun.*, vol. COM-31, Mar. 1983.
[65] H. Murakami, S. Matsumoto, Y. Hatori, and H. Yamamoto, "15/30 Mbit/s universal digital TV codec using a median adaptive predictive coding method," *IEEE Trans. Commun.*, vol. COM-35, June 1987.
[66] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial gray scale," *Proc. Society for Information Display*, vol. 17, no. 2, 1976.
[67] B. Girod, H. Almer, L. Bengtsson, B. Christenson, and P. Weiss, "A subjective evaluation of noise-shaping quantization for adaptive intrainterframe DPCM coding of color television signals," *IEEE Trans. Commun.*, vol. COM-36, Mar. 1988.
[68] S. Matsumoto, *et al.*, "120/140 Mbps intrafield DPCM systems transmission," in *Proc. 2nd Intl. Workshop on Signal Processing of HDTV*, L'aquila, Italy, Mar. 1988.
[69] W. H. Chen and W. K. Pratt, "Scene adaptive coder," *IEEE Trans. Commun.*, vol. COM-32, pp. 225–232, Mar. 1984.
[70] R. J. Clarke, *Transform Coding of Images*. Academic Press, 1985.
[71] H. Lohscheller, "A subjectively adapted image communication system," *IEEE Trans. Commun.*, vol. COM-32, pp. 1316–1322, Dec. 1984.
[72] CCIR-CMTT/2, "Digital transmission of component-coded television signals at 30–34 Mbits/s and 45 Mbits/s using the discrete cosine transform," *Document CMTT/2-66*, July 1988.
[73] C. T. Chen and D. J. Le Gall, "A *K*th order adaptive transform coding algorithm for image data compression," *Proc. SPIE Intl. Symp.*, San Diego, CA, Aug. 1989.
[74] ISO—IEC/JTC1/SC2/WG8 Document N640 R1, "Adaptive discrete cosine transform coding scheme for still image telecommunication services."
[75] A. Artieri, S. Kritter, F. Jutand, and N. Demassieux, "A One chip VLSI for real time two-dimensional discrete cosine transform," *Proc. ISCAS 88*, June 1988.
[76] Telettra (Italy), "Algorithm for broadcast quality encoding of NTSC television for transmission at 44.736 Mbits/s (DS3)," submitted to ANSI committee TI-Y1.1, USA, July 1987.
[77] P. J. Burt and E. H. Adelson, "The Laplacian Pyramid as a compact image code," *IEEE Trans. Commun.*, vol. COM-31, Apr. 1983.
[78] M. Vetterli, "Multi-dimensional sub-band coding: Some theory and algorithms," *Signal Processing*, vol. 6, pp. 97–112, 1984.
[79] J. W. Woods and S. D. O'Neil, "Sub-band coding of images," *IEEE Trans. ASSP*, vol. ASSP-34, Oct. 1986.
[80] H. Gharavi and A. Tabatabai, "Sub-band coding of digital images using two-dimensional quadrature mirror filtering," *Proc. SPIE*, vol. 707, pp. 51–61, Sept. 1986.
[81] D. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," *Proc. ICASSP 88*, Apr. 1988.
[82] D. Le Gall, H. Gaggioni and C. T. Chen, "Transmission of HDTV signals under 140 Mbps using a sub-band decomposition and the discrete cosine transform," in *Proc. 2nd Intl. Workshop on Signal Processing of HDTV*, L'Aquila, Italy, 1988.
[83] H. Malvar, "The LOT: A link between block transform coding and multirate filter bank," *Proc. IEEE Intl. Symp. on Circuits and Systems*, Helsinki, June 1988.
[84] M. Vetterli and D. Le Gall, "Perfect reconstruction filter banks: Some properties and factorizations," *IEEE Trans. ASSP*, vol. 37, July 1989.
[85] T. Kronander, "Some aspects of perception based image coding," Ph.D. Thesis, University of Linkoping, 1989.
[86] T. Kondo: "Adaptive dynamic range coding scheme," *PCS 1986*, Tokyo, Japan.
[87] R. Baker and R. Gray, "Image compression using non-adaptive spatial vector quantization," *Proc. 16th Asilomar Conf. on Circuits, Systems and Computers*, pp. 55–61, 1982.
[88] R. Aravind and A. Gersho, "Low-rate image coding with finite state vector quantization," *Proc. ICASSP 1986*.
[89] J. G. Bernstein, S. Ericson, and B. L. Hinman, "A programmable architecture for low bit rate video coding," *Proc. Intl. Workshop on 64 Kbits/s Coding of Moving Video*, Hannover, June 1988.
[90] P. Strobach, D. Schutt, and W. Tengler, "Space variant regular decomposition quadtrees in adaptive inter-frame coding," *Proc. ICASSP '88*, New York, Apr. 1988.
[91] J. Vaisey and A. Gersho, "Variable rate image coding using quad-trees and vector quantization," *Signal Processing IV*, 1988, pp. 1133–1136.
[92] R. M. Gray, "Vector quantization," *IEEE ASSP Magazine*, pp. 4–29, Apr. 1984.
[93] A. N. Netravali and J. D. Robbins, "Motion compensated television coding—Part I," *Bell Syst. Tech. J.*, vol. 58, Mar. 1979.
[94] C. Caffario and F. Rocca, "The differential methods for image motion estimation," in *Image Sequence Processing and*

Dynamic Scene Analysis, T. S. Huang, Ed. Berlin: Springer-Verlag 1983. September 1976.
[95] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, Dec. 1981.
[96] Y. Ninomiya and Y. Ohtsuka, "A motion-compensated interframe coding scheme for television pictures," *IEEE Trans. Commun.*, vol. COM-30, Jan. 1982.
[97] S. Okubo, R. Nicol, B. Haskell and S. Sabri, "Progress of CCITT standardization on *n**384 kbits/s video codec," *Proc. IEEE GLOBECOM*, Tokyo, 1987.
[98] M. Bierling, "Displacement estimation by hierarchical block matching," *3rd SPIE Symp. on Visual Communications*, Nov. 1988, Cambridge, Massachussets.
[99] K. M. Yang, L. Wu, H. Chong, and M. T. Sun, "VLSI implementation of motion compensation full-search block matching algorithm," *SPIE Visual Communications and Image Processing*, vol. 1001, Cambridge, MA, Nov. 1988.



Sharad Singhal (Member, IEEE) received the B.S. degree in electrical engineering from the Indian Institute of Technology, Kanpur, India, in 1977 and the M.S. and Ph.D. degrees from Yale University, New Haven, CT, in 1978 and 1982 respectively, both in electrical engineering. In 1982 he joined the Acoustics Research Department at AT&T Bell Laboratories in Murray Hill, where he worked on speech coding. He joined Bell Communications Research, Inc. in 1984, where he is currently district manager of the Visual Communications Research Group. His current research interests are in the areas of speech and audio coding, speech recognition, digital signal processing and packet video.

Dr. Singhal is a member of the Acoustical Society of America.

Didier Le Gall was born in Paris, France in 1954. He received the Diplome d'ingenieur from Ecole Centrale de Lyon, Ecully, France in 1976, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at Los Angeles (UCLA) in 1977 and 1981, respectively. From 1982 to 1985 he was with the medical imaging division of Thomson CSF, Paris, France, where he pursued research in reconstruction and display algorithm for computerized tomography. In 1985, he joined Bell Communications Research, Morristown, New Jersey, first as a Member of Technical Staff working on signal processing applied to image communications, then as a District Manager of the Visual Communications Group. His research interests lie in the field of signal processing, filter banks, digital image compression as well as high definition television. Dr. Le Gall was also an adjunct professor at the Department of Electrical Engineering, Columbia University, New York. In February 1990, he joined C-Cube Microsystem, San Jose, California, where he is currently Director of Research.

Cheng-Tie Chen received the B.S. degree from the National Taiwan University in 1977, and M.S.E. and Ph.D. degrees from the University of Pennsylvania in 1981 and 1983, respectively, all in Electrical Engineering. From 1983 to 1984, he was employed, as a Research Associate, at the Department of Electrical Engineering, Princeton University, where he was engaged in the research of non-Gaussian and sensor-array signal processings. From 1984 to 1987, he was working for the Research Laboratories, Eastman Kodak Company, where he was a Senior Research Scientist. His research activities there were in the areas of image restoration and image coding/transmission. Since 1987, he has been with Bell Communications Research in Morristown NJ, as a Member of Technical Staff. His current research interests are in image/video coding and packet video. Dr. Chen was also an adjunct associate professor at the Department of Electrical Engineering, University of Pennsylvania in 1988.