6.1

Basic Concepts and Techniques of Video Coding and the H.261 Standard

Barry Barnett
IBM Corporation

1 Introduction

The subject of video coding is of fundamental importance to many areas in engineering and the natural and perceptual sciences. Video engineering is quickly becoming a largely digital discipline although analog TV transmission is still by far the mainstay around the world. Digital transmission of television signals via satellites is commonplace, and widespread HDTV terrestrial transmission began in 2000 and is targeted to become the default transmission standard in the United States by 2006. Video compression is an absolute requirement for the growth and success for the low bandwidth transmission and storage of digital video signals. Video encoding is used wherever digital video communications, storage, processing, acquisition, and reproduction occur. The transmission of high quality multimedia information over high speed computer networks is a central problem in the design of *quality of services* (QoS) for digital transmission providers. The *Motion Pictures Expert Group* (MPEG) finalized four well known encoding standards, MPEG-1, MPEG-2, MPEG-4, and MPEG-7. MPEG-1 and MPEG-2 define methods for the transmission of digital video information for multimedia and television formats. The MPEG-4 standard, which was adopted in 1999, specifically addresses the transmission of very low bit rate video by introducing the notion of media objects which are made up of audio, visual, or audiovisual content. It is targeted to satisfy the needs of audiovisual content authors, service providers, and end users. The MPEG-7 standard, which was adopted in 2001, defines audiovisual content storage and retrieval services (Sections 9.1 and 9.2 discuss video storage and retrieval). An aspect central to each of the MPEG standards are the video encoding and decoding algorithms that make digital video applications practical. The MPEG standards are discussed in Sections 6.4 and 6.5.

Video compression not only reduces the storage requirements or transmission bandwidth of digital video applications, but also affects many system performance tradeoffs. The design and selection of a video encoder therefore is not only based on its ability to compress information. Issues such as bit rate versus distortion criteria, algorithm complexity, transmission channel characteristics, algorithm symmetry

versus asymmetry, video source statistics, fixed versus variable rate coding, and standards compatibility should be considered in order to make good encoder design decisions.

The growth of digital video applications and technology in the last few years has been explosive, and video compression is playing a central role in this success. Yet, the video coding discipline is relatively young and certainly will evolve and change significantly over the next few years. For instance, perceptual based video encoding research is still relatively young and promises to be able to significantly influence the future course of the discipline. Research in video coding has great vitality and the body of work is significant. It is apparent that this relevant and important topic will have an immense affect on the future of digital video technologies.

2 Introduction to Video Compression

Video or visual communications require significant amounts of information transmission. Video compression as considered here, involves the bit rate reduction of the digital video signal carrying visual information. Traditional video based compression like other information compression techniques, focuses on eliminating redundancy and unimportant elements of the source. The degree to which the encoder reduces the bit rate is called its *coding efficiency*, or equivalently its inverse is termed the *compression ratio*, i.e.,

$$\begin{aligned}\text{coding efficiency} &= (\text{compression ratio})^{-1} \\ &= \text{encoded bit rate/decoded bit rate.}\end{aligned} \tag{1}$$

Compression can be a lossless or lossy operation. Due to the immense volume of video information, lossy operations are a key element used in video compression algorithms. The loss of information or distortion measure is usually evaluated using the *mean square error* (MSE), *mean absolute error* (MAE) criteria, or *peak signal-to-reconstruction noise* (PSNR),

$$\begin{aligned}MSE &= \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j)-\hat{I}(i,j)]^2 \\ MAE &= \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|I(i,j)-\hat{I}(i,j)\right| \\ PSNR &= 20\log_{10}\left(\frac{2^n}{MSE^{1/2}}\right),\end{aligned} \tag{2}$$

for image I and its reconstructed image $\hat{I}$, pixel indices $1 \le i \le M$ and $1 \le j \le N$, image size $N \times M$ pixels, and n bits per pixel. The MSE, MAE, and PSNR as described here are global measures and do not necessarily give a good indication of the reconstructed image quality. In the final analysis, the human observer determines the quality of the reconstructed image and video quality. The concept of distortion versus coding efficiency is one of the most fundamental tradeoffs in the technical evaluation of video encoders. The topic of perceptual quality assessment of compressed images and video is discussed in Section 8.2.

Video signals contain information in three dimensions. These dimensions are modeled as *spatial* and *temporal* domains for video encoding purposes. Digital video compression methods generally seek to minimize information redundancy independently in each domain. The major international video compression standards (MPEG-1, MPEG-2, MPEG-4, H.261, H.262, and H.263) use this approach. Figure 1 schematically depicts a generalized video compression system that implements the spatial and temporal encoding of a digital image sequence. Each image in the

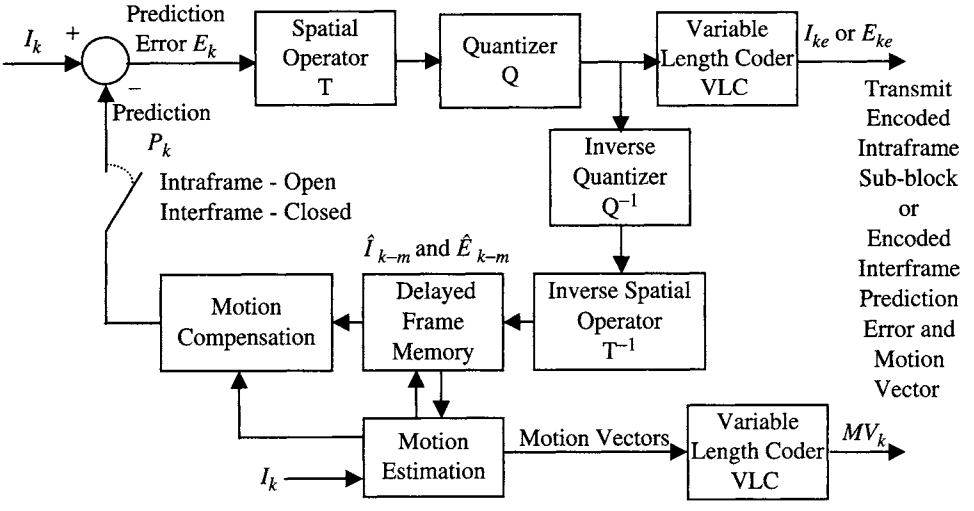


FIGURE 1 Generalized video compression system.

sequence I_k is defined as in Eq. (2). The spatial encoder operates on image *blocks*, typically on the order of 8×8 pixels each. The temporal encoder generally operates on 16×16 pixel image blocks. The system is designed for two modes of operation, the *intraframe* mode and the *interframe* mode.

The single layer feedback structure of this generalized model is representative of the encoders that are recommended by the *International Standards Organization* (ISO) and *International Telecommunications Union* (ITU) video coding standards MPEG-1, MPEG-2/H.262, MPEG-4, and H.261 [1, 2, 3, 4]. The feedback loop is used in the Interframe mode of operation and generates a *prediction error* between the image blocks of the current frame and the current prediction frame. The prediction is generated by the *motion compensator*. The *motion estimation* unit creates *motion vectors* for each 16×16 block. The motion vectors and previously reconstructed frame are fed to the motion compensator to create the prediction.

The intraframe mode spatially encodes an entire current frame on a periodic basis, such as every 15 frames, to ensure that systematic errors do not continuously propagate. Intraframe mode will also be used to spatially encode a block whenever the Interframe encoding mode cannot meet its performance threshold. The intraframe versus interframe mode selection algorithm is not included in this diagram. It is responsible for controlling the selection of the encoding functions, data flows, and output data streams for each mode.

The intraframe encoding mode does not receive any input from the feedback loop. I_k is spatially encoded, and subsequently encoded by the VLC forming I_{ke}, which is transmitted to the decoder. The receiver decodes I_{ke} producing the reconstructed image sub-block $\hat{I}_k$. During the interframe coding mode, the current frame prediction P_k is subtracted from the current frame input I_k to form the current prediction error E_k. The prediction error is then spatially and VLC encoded to form E_{ke} and is transmitted along with the VLC encoded motion vectors MV_k. The decoder can reconstruct the current frame $\hat{I}_k$ using the previously reconstructed frame $\hat{I}_{k-1}$(stored in the decoder), the current frame motion vectors, and the prediction error. The motions vectors MV_k operate on $\hat{I}_{k-1}$ to generate the current prediction frame P_k. The encoded prediction error E_{ke} is decoded to produce the reconstructed prediction error $\hat{E}_k$. The prediction error is added to the prediction to form the current frame $\hat{I}_k$. The functional elements of the generalized model are described here in detail.

- **Spatial operator** — This element is generally a unitary two dimensional linear transform, but in principle can be any unitary operator that can distribute most of the signal energy into a small number of coefficients, i.e., decorrelate the signal data. Spatial transformations are successively applied to small image blocks in order to take advantage of the high degree of data correlation in adjacent image pixels. The most widely used spatial operator for image and video coding is the *discrete cosine transform* (DCT). It is applied to 8×8 pixel image blocks, and is well suited for image transformations because it uses real computations with fast implementations, provides excellent decorrelation of signal components, and avoids generation of spurious components between the edges of adjacent image blocks.
- **Quantizer** — The spatial or transform operator is applied to the input in order to arrange the signal into a more suitable format for subsequent lossy and lossless coding operations. The quantizer operates on the transform generated coefficients. This is a lossy operation that can result in a significant reduction in the bit rate. The quantization method used in this kind of video encoder is usually scalar and non-uniform. The scalar quantizer simplifies the complexity of the operation as compared to *vector quantization* (VQ). The non-uniform quantization interval is sized according to the distribution of the transform coefficients in order to minimize the bit rate and the distortion created by the quantization process. Alternatively, the quantization interval size can be adjusted based on the performance of the *human visual system* (HVS). The *Joint Pictures Expert Group* (JPEG) standard includes two (luminance and color difference) HVS sensitivity weighted quantization matrices in its "Examples and Guidelines" annex. JPEG coding is discussed in Sections 5.5 and 5.6.
- **Variable length coding** — The lossless variable length coding (VLC) is used to effectively exploit the "symbolic" redundancies contained in each block of quantized transform coefficients. This step is termed "entropy coding" to designate that the encoder is designed to minimize the source entropy. The VLC is applied to a serial bit stream that is generated by scanning the transform coefficient block. The scanning pattern should be chosen with the objective of maximizing the performance of the VLC. The MPEG encoder for instance, describes a zigzag scanning pattern that is intended to maximize transform zero coefficient run lengths. The H.261 VLC is designed to encode these run lengths using a variable length *Huffman* code.

The feedback loop sequentially reconstructs the encoded spatial and prediction error frames and stores the results in order to create a prediction for the current image sub-block. The elements required to do this are the inverse quantizer, inverse spatial operator, delayed frame memory, motion estimator, and motion compensator.

- **Inverse operators** — The inverse operators Q^{-1} and T^{-1} are applied to the encoded current frame I_{ke} or the current prediction error E_{ke} in order to reconstruct and store the encoded frame for the motion estimator and

motion compensator to generate the next prediction frame.

- **Delayed frame memory** — Both current and previous frames must be available to the motion estimator and motion compensator to generate a prediction frame. The number of previous frames stored in memory can vary based upon the requirements of the encoding algorithm. MPEG-1 defines a B-frame which is a bidirectional encoding that requires that motion prediction be performed in both the forward and backward directions. This necessitates storage of multiple frames in memory.
- **Motion estimation** — The temporal encoding aspect of this system relies on the assumption that rigid body motion is responsible for the differences between two or more successive frames. The objective of the motion estimator is to estimate the rigid body motion between two frames. The motion estimator operates on all current frame 16×16 image blocks and generates the pixel displacement or *motion vector* for each block. The technique used to generate motion vectors is called *block-matching motion estimation* and is discussed further in Section 5.4. The method uses the current frame I_k and the previous reconstructed frame $\hat{I}_{k-1}$ as input. Each block in the previous frame is assumed to have a displacement that can be found by searching for it in the current frame. The search is usually constrained to be within a reasonable neighborhood so as to minimize the complexity of the operation. Search matching is usually based on a minimum MSE or MAE criteria. When a match is found, the pixel displacement is used to encode the particular block. If a search does not meet a minimum MSE or MAE threshold criteria, the motion compensator will indicate that the current block is to be spatially encoded using Intraframe mode.
- **Motion compensation** — The motion compensator makes use of the current frame motion estimates MV_k and the previously reconstructed frame $\hat{I}_{k-1}$ to generate the current frame prediction P_k. The current frame prediction is constructed by placing the previous frame blocks into the current frame according to the motion estimate pixel displacement. The motion compensator uses the threshold criteria to decide which blocks will be encoded as prediction error blocks using motion vectors and which blocks will only be spatially encoded.

The generalized model does not address some video compression system details such as the bit-stream syntax (which supports different application requirements), or the specifics of the encoding algorithms. These issues are dependent upon the video compression system design.

Alternative video encoding models have also been the focus of current research. Three-dimensional (3D) video information can be compressed directly using VQ or 3D *wavelet* encoding models. VQ encodes a 3D block of pixels as a codebook index that denotes its "closest or nearest neighbor" in the minimum squared or absolute error sense. However, the VQ codebook size grows on the order as the number of possible inputs. Searching the codebook space for the nearest neighbor is generally very computationally complex, but structured search techniques can provide good bit rates, quality, and computational performance. *Tree-structured* VQ (TSVQ) [14] reduces the search complexity from codebook size N to $\log N$, with a corresponding loss in average distortion. The simplicity of the VQ decoder (it only requires a simple table lookup for the transmitted codebook index), and its bit rate-distortion performance make it an attractive alternative for specialized applications. The complexity of the codebook search generally limits the use of VQ in real-time applications. VQ quantizers have also been proposed for Interframe, variable bit rate, and subband video compression methods [5].

Three dimensional wavelet encoding is a topic of recent interest. This video encoding method is based on the *discrete wavelet transform* methods discussed in Section 5.4. The wavelet transform decomposes a bandwidth limited signal into a *multiresolution* representation. The multiresolution decomposition makes the wavelet transform an excellent signal analysis tool because signal characteristics can be viewed in a variety of time-frequency or space-frequency scales. The wavelet transform is implemented in practice via the use of multiresolution or *subband* filterbanks [6]. The wavelet filterbank is well suited for video encoding because of its ability to adapt to the multiresolution characteristics of video signals. Wavelet transform encodings are naturally hierarchic in their time-frequency representation and easily adaptable for *progressive transmission* [7]. They have also been shown to possess excellent bit rate-distortion characteristics.

Direct three dimensional video compression systems suffer from a major drawback for real-time encoding and transmission. In order to encode a sequence of images in one operation, the sequence must be buffered. This introduces a buffering and computational delay that can be very noticeable in the case of real-time video communications.

Video compression techniques treating visual information in accordance with *human visual system* models have recently been introduced. These methods are termed "second-generation or object-based," and attempt to achieve very large compression ratios by imitating the operations of the HVS. The HVS model has also been incorporated into more traditional video compression techniques by reflecting visual perception into various aspects of the coding algorithm. HVS weightings have been designed for the DCT AC coefficients quantizer used in the MPEG encoder. A discussion of these techniques can be found in Section 6.3.

Digital video compression is currently enjoying tremendous growth in part due to the great advances in VLSI, ASIC, and microcomputer technology in the last decade. The real-time nature of video communications necessitates the use of general purpose and specialized high performance hardware devices. In the near future, advances in design and manufacturing technologies will create hardware devices that will allow greater adaptability, interactivity, and interoperability of video applications. For instance, MPEG-7 has defined format free operations for the storage and retrieval of audio-visual information that is being used by digital cable TV vendors for "on demand" delivery of digital video content.

3 Video Compression Application Requirements

A wide variety of digital video applications currently exist. They range from simple low resolution and bandwidth applications (multimedia, PicturePhone) to very high resolution and bandwidth (HDTV) demands. This section will present requirements of current and future digital video applications and the demands they place on the video compression system.

To demonstrate the importance of video compression, the transmission of digital video television signals is presented. The bandwidth required by a digital television signal is approximately one-half the number of picture elements (pixels) displayed per second. The analog video monitor pixel size in the vertical dimension is the distance between scanning lines, and the horizontal dimension is the distance the scanning spot moves during ½ cycle of the highest video signal transmission frequency. The video signal bandwidth is given by Eq. (3),

$$\begin{aligned} B_W &= (\text{cycles/frame})(F_R) \\ &= (\text{cycles/line})(N_L)(F_R) \\ &= \frac{(0.5)(\text{aspect ratio})(F_R)(N_L)(R_H)}{0.84} \\ &= (0.8)(F_R)(N_L)(R_H) \end{aligned} \tag{3}$$

where

B_W = video signal system bandwidth
F_R = number of frames transmitted per second (fps)
N_L = number of scanning lines per frame
R_H = horizontal resolution (lines), proportional to pixel resolution

The NTSC picture aspect ratio is 4/3, the constant 0.5 is the ratio of the number of cycles to the number of lines, and the factor 0.84 is the fraction of the horizontal scanning interval that is devoted to signal transmission.

The *National Television Systems Committee* (NTSC) transmission standard used for television broadcasts in the United States has the following parameter values,

$$F_R = 29.97\,\text{fps}$$

$$N_L = 525\,\text{lines}$$

$$R_H = 340\,\text{lines}.$$

This yields an analog video system bandwidth B_W of 4.2 MHz for the NTSC broadcast system. In order to transmit a color digital video signal, the digital pixel format must be defined. The digital color pixel is made of three components: One luminance (Y) component occupying 8 bits, and two color difference components (U and V) each requiring 8 bits. The NTSC picture frame has $720 \times 480 \times 2$ total luminance and color pixels. In order to transmit this information for an NTSC broadcast system at 29.97 frames per second, the following bandwidth is required,

$$\begin{aligned} \text{Digital BW} \simeq {}^1/_2\,\text{bitrate} &= {}^1/_2(29.97\,\text{fps}) \times (24\,\text{bits/pixel}) \\ &\quad \times (720 \times 480 \times 2\,\text{pixels/frame}) \\ &= 249\ \text{MHz}. \end{aligned}$$

This represents an increase of approximately 59 times the required NTSC system bandwidth, and about 41 times the full transmission channel bandwidth (6 MHz) for current NTSC signals. HDTV picture resolution requires up to three times more raw bandwidth than this example! (Two transmission channels totaling 12 MHz are allocated for terrestrial HDTV transmissions.) It is clear from this example that terrestrial television broadcast systems have to use digital transmission and digital video compression to achieve the overall bit rate reduction and image quality required for HDTV signals.

The example not only points out the significant system bandwidth requirements for digital video information, but also indirectly brings up the issue of digital video quality requirements. The tradeoff between bit rate and quality or distortion is a fundamental issue facing the design of video compression systems. To this end, it is important to fully characterize an application's video communications requirements before designing or selecting an appropriate video compression system. Factors that should be considered in the design and selection of a video compression system include the following items:

- **Video characteristics** — Video parameters such as the dynamic range, source statistics, pixel resolution, and noise content can affect the performance of the compression system.

- **Transmission requirements** — Transmission bit rate requirements determine the power of the compression system. Very high transmission bandwidth, storage capacity, or quality requirements may necessitate lossless compression. Conversely, extremely low bit rate requirements may dictate compression systems that trade-off image quality for a large compression ratio. *Progressive transmission* is a key issue for selection of the compression system. It is generally used when the transmission bandwidth exceeds the compressed video bandwidth. Progressive coding refers to a multi-resolution, hierarchic, or subband encoding of the video information. It allows for transmission and reconstruction of each resolution independently from low to high resolution. Channel errors affect system performance and the quality of the reconstructed video. Channel errors can affect the bit stream randomly or in burst fashion. The channel error characteristics can have different effects on different encoders, and can range from local to global anomalies. In general, transmission error correction codes (ECC) are used to mitigate the effect of channel errors, but awareness and knowledge of this issue is important.
- **Compression system characteristics and performance** — The nature of video applications makes many demands on the video compression system. Interactive video applications such as videoconferencing demand that the video compression systems have symmetric capabilities. That is, each participant in the interactive video session must have the same video encoding and decoding capabilities, and that the system performance requirements must be met by both the encoder and decoder. On the other hand, television broadcast video has significantly greater performance requirements at the transmitter because it has the responsibility of providing real-time high quality compressed video that meets the transmission channel capacity. Digital video system implementation requirements can vary significantly. Desktop televideo conferencing can be implemented using software encoding and decoding or may require specialized hardware and transmission capabilities to provide high quality performance. The characteristics of the application will dictate the suitability of the video compression algorithm for particular system implementations. The importance of the encoder and system implementation decision cannot be overstated, system architectures and performance capabilities are changing at a rapid pace and the choice of the best solution requires careful analysis of the all possible system and encoder alternatives.
- **Rate-Distortion requirements** — The rate-distortion requirement is a basic consideration in the selection of the video encoder. The video encoder must be able to provide the bit rate(s) and video fidelity (or range of video fidelity) required by the application. Otherwise, any aspect of the system may not meet specifications. For example, if the bit rate specification is exceeded in order to support a lower MSE, a larger than expected transmission error rate may cause a catastrophic system failure.
- **Standards requirements** — Video encoder compatibility with existing and future standards is an important consideration if the digital video system is required to inter-operate with existing and/or future systems. A good example is that of a desktop videoconferencing application supporting a number of legacy video compression standards. This requires support of the older video encoding standards on new equipment designed for a newer incompatible standard. Videoconferencing equipment not supporting the old standards would not be capable or as capable to work in environments supporting older standards.

These factors are displayed in Table 1 to demonstrate video compression system requirements for some common video communications applications. The video compression system designer as a minimum should consider these factors in making a determination about the choice of video encoding algorithms and technology to implement.

4 Digital Video Signals and Formats

Video compression techniques make use of signal models in order to be able to utilize the body of digital signal analysis/processing theory and techniques that have been developed over the past fifty or so years. The design of a video compression system as represented by the generalized model introduced in Section 2, requires knowledge of the signal characteristics, and the digital processes that are used to create the digital video signal. It is also highly desirable to understand video display systems, and the behavior of the HVS.

4.1 Sampling of Analog Video Signals

Digital video information is generated by sampling the *intensity* of the original continuous analog video signal $I(x, y, t)$ in three dimensions. The spatial component of the video signal is sampled in the horizontal and vertical dimensions (x, y), and the temporal component is sampled in the time dimension (t). This generates a series of digital images or image sequence $I(i, j, k)$. Video signals that contain colorized information are usually decomposed into three parameters (YC_rC_b, YUV, RGB, etc.) whose intensities are likewise sampled in three dimensions. The sampling process inherently quantizes the video signal due to the digital word precision used to represent the intensity values. Therefore the original analog signal can never be reproduced exactly,

TABLE 1 Digital video application requirements

Application	Bit rate Req.	Distortion Req.	Transmission Req.	Computational Req.	Standards Req.
Network video on demand	1.5 Mbps 10 Mbps	High medium	Internet 100 Mbps lan	MPEG-1 MPEG-2/4	MPEG-1 MPEG-2/4 MPEG-7
Video phone	64 Kbps	High distortion	ISDN p × 64	H.261 encoder h.261 decoder	H.261
Desktop multimedia video CDROM	1.5 Mbps	High distortion to medium	PC channel	MPEG-1 decoder	MPEG-1 MPEG-2 MPEG-7
Desktop LAN videoconference	10 Mbps	Medium distortion	Fast ethernet 100 Mbps	Hardware decoders	MPEG-2/4, H.261
Desktop WAN videoconference	1.5 Mbps	High distortion	Ethernet	Hardware decoders	MPEG-1, MPEG-4, H.263
Desktop dial-up videoconference	64 Kbps	Very high distortion	POTS and internet	Software decoder	MPEG-4, H.263
Digital satellite television	10 Mbps	Low distortion	Fixed service satellites	MPEG-2 decoder	MPEG-2
HDTV	20 Mbps	Low distortion	12 MHz terrestrial link	MPEG-2 decoder	MPEG-2
DVD	20 Mbps	Low distortion	PC channel	MPEG-2 decoder	MPEG-2

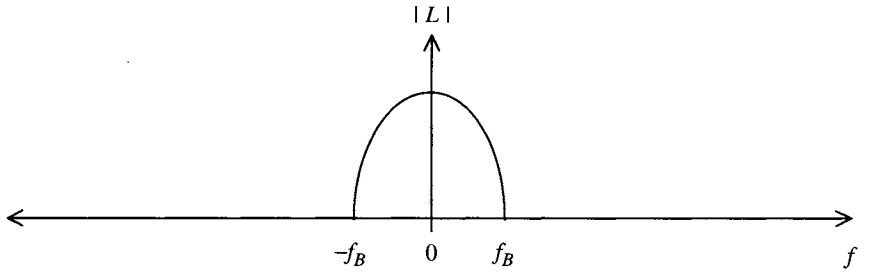
but for all intents and purposes, a high-quality digital video representation can be reproduced with arbitrary closeness to the original analog video signal. The topic of video sampling and interpolation is discussed in Section 7.2.

An important result of sampling theory is the *Nyquist sampling theorem.* This theorem defines the conditions under which sampled analog signals can be "perfectly" reconstructed. If these conditions are not met, the resulting digital signal will contain *aliased* components which introduce artifacts into the reconstruction. The Nyquist conditions are depicted graphically for the one dimensional case in Fig. 2.

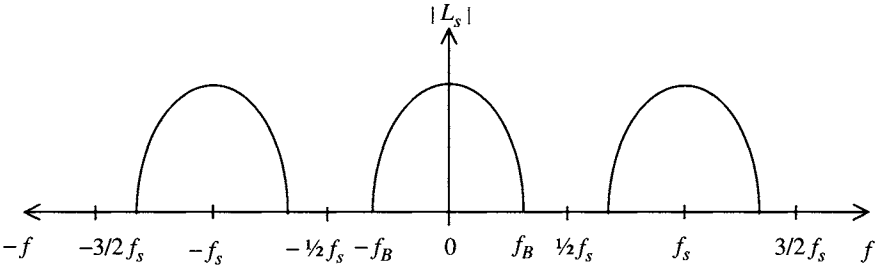
The one dimensional signal l is sampled at rate f_s. It is band limited (as are all real world signals) in the frequency domain with an upper frequency bound of f_B. According to the Nyquist sampling theorem, if a bandlimited signal is sampled, the resulting *Fourier* spectrum is made up of the original signal spectrum $| L |$ plus replicates of the original spectrum spaced at integer multiples of the sampling frequency f_s. Diagram (a) in Fig. 2 depicts the magnitude $| L |$ of the Fourier spectrum for l. The magnitude of the *Fourier* spectrum $| L_s |$ for the sampled signal l_s is shown for two cases. Diagram (b) presents the case where the original signal l can be reconstructed by recovering the central spectral island. Diagram (c) displays the case where the Nyquist sampling criteria has not been met and spectral overlap occurs. The spectral overlap is termed *aliasing* and occurs when $f_s < 2f_B$. When $f_s > 2f_B$, the original signal can be reconstructed by using a low pass digital filter whose pass band is designed to recover $| L |$. These relationships provide a basic framework for the analysis and design of digital signal processing systems.

Two-dimensional or spatial sampling is a simple extension of the one-dimensional case. The Nyquist criteria has to be obeyed in both dimensions, i.e., the sampling rate in the horizontal direction must be two times greater than the upper frequency bound in the horizontal direction, and the sampling rate in the vertical direction must be two times greater than the upper frequency bound in the vertical direction. In practice, spatial sampling grids are square so that an equal number of samples per unit length in each direction are collected. *Charge coupled devices* (CCD) are typically used to spatially sample analog imagery and video. The sampling grid spacing of these devices is more than sufficient to meet the Nyquist criteria for most resolution and application requirements. The electrical characteristics of CCDs have a greater affect on the image or video quality than its sampling grid size.

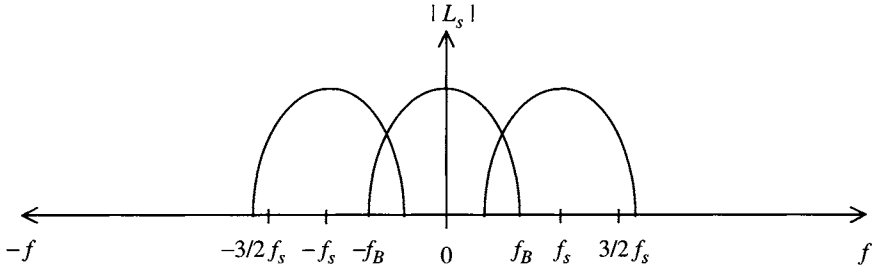
Temporal sampling of video signals is accomplished by capturing a spatial or image frame in the time dimension. The temporal samples are captured at a uniform rate of about 60 fields per second for NTSC television and 24 fps for a motion film recording. These sampling rates are significantly less than the spatial sampling rate. The maximum temporal frequency that can be reconstructed according to the Nyquist frequency criteria is 30 Hz in the case of television broadcast. Therefore any rapid intensity change (caused for instance by a moving edge) between two successive frames will cause aliasing because the harmonic frequency content of such a step-like function exceeds the Nyquist frequency. Temporal aliasing of this kind can be greatly mitigated in CCDs by the use of low pass temporal filtering to remove the high frequency content. *Photoconductor storage tubes* are used for



(a) Magnitude of Fourier Spectrum for input l



(b) Magnitude of Fourier Spectrum for sampled input l_s, with $f_s > 2f_B$



(c) Magnitude of Fourier Spectrum for sampled input l_s, with $f_s < 2f_B$

FIGURE 2 Nyquist sampling theorem.

recording broadcast television signals. They are analog scanning devices whose electrical characteristics filter the high frequency temporal content and minimize temporal aliasing. Indeed, motion picture film also introduces low pass filtering when capturing image frames. The exposure speed and the response speed of the photo chemical film combine to mitigate high frequency content and temporal aliasing. These factors cannot completely stop temporal aliasing, so intelligent use of video recording devices is still warranted, e.g., the main reason movie camera panning is done very slowly is to minimize temporal aliasing.

In many cases where fast motions or moving edges are not well resolved due to temporal aliasing, the human visual system will interpolate such motion and provide its own perceived reconstruction. The HVS is very tolerant of temporal aliasing by using its own knowledge of natural motion to provide motion estimation and compensation to the image sequences generated by temporal sampling. The combination of temporal filtering in sampling systems and the mechanisms of human visual perception, reduce the affects of temporal aliasing such that temporal under sampling (sub-Nyquist sampling) is acceptable in the generation of typical image sequences intended for general purpose use.

4.2 Digital Video Formats

Sampling is the process used to create the image sequences used for video and digital video applications. Spatial sampling and quantization of a natural video signal digitizes the image plane into a two dimensional set of digital pixels that define a digital image. Temporal sampling of a natural video signal creates a sequence image frames typically used for motion pictures and television. The combination of spatial and temporal sampling creates a sequence of digital images termed digital video. As described earlier, the digital video signal intensity is defined as $I(i, j, k)$, where $0 \leq i \leq M$,

TABLE 2 Digital composite television parameters

Description	NTSC	PAL
Analog video bandwidth (MHz)	4.2	5.0
Aspect ratio, hor Size/vert Size	4/3	4/3
Frames per second	29.97	25
Lines per second	525	625
Interlace ratio, fields:frames	2:1	2:1
Subcarrier frequency (MHz)	3.58	4.43
Sampling frequency (MHz)	14.4	17.7
Samples per active line	757	939
Bit rate (Mbps)	114.5	141.9

$0 \leq j \leq N$ are the horizontal and vertical spatial coordinates, and $0 \leq k$ is the temporal coordinate.

The standard digital video formats introduced here are used in the broadcast for both analog and digital television, as well as computer video applications. Composite television signal digital broadcasting formats are included here due to their use in video compression standards, digital broadcasting, and standards format conversion applications. Knowledge of these digital video formats provides background for understanding the international video compression standards developed by the *International Telecommunications Union* (ITU) and the *International Standards Organization* (ISO). These standards contain specific recommendations for use of the digital video formats described here.

Composite television digital video formats are used for the digital broadcasting, SMPTE digital recording, and conversion of television broadcasting formats. Table 2 contains both analog and digital system parameters for the *National Television Systems Committee* (NTSC), and *phase alternating lines* (PAL) composite broadcast formats.

Component television signal digital video formats have been defined by the *International Consultative Committee for Radio* (CCIR) Recommendation 601. It is based on component video with one luminance (Y) and two color difference signals (C_r and C_b). The raw bit rate for the CCIR 601 format is 162 Mbps. Table 3 contains important systems parameters of the CCIR 601 digital video studio component recommendation for both NTSC and PAL/SECAM (*sequentiel couleur avec memoire*).

TABLE 3 Digital video component television parameters for CCIR 601

Description	NTSC	PAL/SECAM
Luminance channel		
Analog video bandwidth (MHz)	5.5	5.5
Sampling frequency (MHz)	13.5	13.5
Samples per active line	710	716
Bit rate (Mbps)	108	108
Two color difference channels		
Analog video bandwidth (MHz)	2.2	2.2
Sampling frequency (MHz)	6.75	6.75
Samples per active line	355	358
Bit rate (Mbps)	54	54

TABLE 4 SIF, CIF, and QCIF digital video formats

Description	SIF NTSC/PAL	CIF	QCIF
Horizontal resolution (Y) pixels	352	360(352)	180(176)
Vertical resolution (Y) pixels	240/288	288	144
Horizontal resolution (C_r, C_b) pixels	176	180(176)	90(88)
Vertical resolution (C_r, C_b) pixels	120/144	144	72
Bits per pixel (bpp)	8	8	8
Interlace fields:frames	1:1	1:1	1:1
Frame rate (fps)	30	30, 15, 10, 7.5	30, 15, 10, 7.5
Aspect ratio hor Size/vert Size	4:3	4:3	4:3
Bit rate (Y) Mbps @ 30 fps	20.3	24.9	6.2
Bitrate (U, V) Mbps @ 30 fps	10.1	12.4	3.1

The ITU Specialist Group (SGXV) has recommended three formats that are used in the ITU H.261, H.263, and ISO MPEG video compression standards. They are the *standard input format* (SIF), *common interchange format* (CIF), and the low bit rate version of CIF called *quarter CIF* (QCIF). Together, these formats describe a comprehensive set of digital video formats that are widely used in current digital video applications. CIF and QCIF support the NTSC and PAL video formats using the same parameters. The SIF format defines different vertical resolution values for NTSC and PAL. The CIF and QCIF formats also support the H.261 modified parameters. The modified parameters are integer multiples of 8 in order to support the 8×8 pixel two-dimensional DCT operation. Table 4 lists this set of digital video standard formats. The modified H.261 parameters are listed in parenthesis.

5 Video Compression Techniques

Video compression systems are generally comprised of two modes that reduce information redundancy in the spatial and the temporal domains. Spatial compression and quantization operates on a single image block, making use of the local image characteristics to reduce the bit rate. The spatial encoder also includes a variable length coder (VLC) inserted after the quantization stage. The VLC stage generates a lossless encoding of the quantized image block. Lossless coding is discussed in Section 5.1. Temporal domain compression

makes use of optical flow models (generally in the form of block-matching motion estimation methods) to identify and mitigate temporal redundancy.

This section presents an overview of some widely accepted encoding techniques used in video compression systems. *Entropy encoders* are lossless encoders that are used in the variable length coding (VLC) stage of a video compression system. They are best used for information sources that are *memoryless* (sources in which each value is independently generated), and try to minimize the bit rate by assigning variable length codes for the input values according to the input *probability density function* (pdf). *Predictive coders* are suited to information sources that have memory, i.e., a source in which each value has a statistical dependency on some number of previous and/or adjacent values. Predictive coders can produce a new source pdf with significantly less statistical variation and entropy than the original. The transformed source can then be fed to a VLC to reduce the bit rate. Entropy and predictive coding are good examples for presenting the basic concepts of statistical coding theory.

Block transformations are the major technique for representing spatial information in a format that is highly conducive to quantization and VLC encoding. Block transforms can provide a coding gain by packing most of the block energy into a fewer number of coefficients. The *quantization* stage of the video encoder is the central factor in determining the rate-distortion characteristics of a video compression system. It quantizes the block transform coefficients according to the bit rate and distortion specifications. Motion compensation takes advantage of the significant information redundancy in the temporal domain by creating current frame predictions based upon block matching motion estimates between the current and previous image frames. Motion compensation generally achieves a significant increase in the video coding efficiency over pure spatial encoding.

5.1 Entropy and Predictive Coding

Entropy coding is an excellent starting point in the discussion of coding techniques because it makes use of many of the basic concepts introduced in the discipline of *information theory* or *statistical communications theory* [8]. The discussion of VLC and predictive coders requires the use of *information source* models to lay the statistical foundation for the development of this class of encoder. An information source can be viewed as a process that generates a sequence of symbols from a finite alphabet. Video sources are generated from a sequence of image blocks that are generated from a "pixel" alphabet. The number of possible pixels that can be generated is 2^n, when n is the number of bits per pixel. The order in which the image symbols are generated depends on how the image block is arranged or scanned into a sequence of symbols. Spatial encoders transform the statistical nature of the original image so that the resulting coefficient matrix can be scanned in a manner such that the resulting source or sequence of symbols contains significantly less information content.

Two useful information sources are used in modeling video encoders; the *discrete memoryless source* (DMS), and *Markov* sources. VLC coding is based on the DMS model, and the predictive coders are based on the Markov source models. The DMS is simply a source in which each symbol is generated independently. The symbols are *statistically independent* and the source is completely defined by its symbols/events and the set of probabilities for the occurrence for each symbol, i.e., $E=\{e_1, e_2, \ldots, e_n\}$ and the set $\{p(e_1), p(e_2), \ldots, p(e_n)\}$, where n is the number of symbols in the alphabet. It is useful to introduce the concept of entropy at this point. Entropy is defined as the average information content of the information source. The information content of a single event or symbol is defined as,

$$I(e_i) = \log \frac{1}{p(e_i)}. \tag{4}$$

The base of the logarithm is determined by the number of states used to represent the information source. Digital information sources use base 2 in order to define the information content using the number of bits per symbol or bit rate. The entropy of a digital source is further defined as the average information content of the source, i.e.,

$$H(E) = \sum_{i=1}^{n} p(e_i) \log_2 \frac{1}{p(e_i)} = -\sum_{i=1}^{n} p(e_i) \log_2 p(e_i) \text{ bits/symbol.} \tag{5}$$

This relationship suggests that the average number of bits per symbol required to represent the information content of the source is the entropy. The *noiseless source coding theorem* states that a source can be encoded with an average number of bits per source symbol that is arbitrarily close to the source entropy. So called entropy encoders seek to find codes that perform close to the entropy of the source. *Huffman* and *arithmetic* encoders are examples of entropy encoders.

Modified Huffman coding [9] is commonly used in the image and video compression standards. It produces good performing variable length codes without significant computational complexity. The traditional Huffman algorithm is a two step process that first creates a table of source symbol probabilities, and then constructs codewords whose lengths grow according to the decreasing probability of a symbol's occurrence. Modified versions of the traditional algorithm are used in the current generation of image and video encoders. The H.261 encoder uses two sets of static Huffman codewords (one each for AC and DC DCT coefficients). A set

of 32 codewords is used for encoding the AC coefficients. The zigzag scanned coefficients are classified according to the zero coefficient run-length and first nonzero coefficient value. A simple table lookup is all that is then required to assign the codeword for each classified pair.

Markov and *random field* source models (discussed in Section 4.3) are well suited to describing the source characteristics of natural images. A Markov source has memory of some number of preceding or adjacent events. In a natural image block, the value of the current pixel is dependent on the values of some the surrounding pixels because they are part of the same object, texture, contour, etc. This can be modeled as an m-th order Markov source, in which the probability of source symbol e_i depends on the last m source symbols. This dependence is expressed as the probability of occurrence of event e_i conditioned on the occurrence of the last m events, i.e., $p(e_i|e_{i-1}, e_{i-2}, \ldots, e_{i-m})$. The Markov source is made up of all possible n^m states, where n is the number of symbols in the alphabet. Each state contains a set of up to n conditional probabilities for the possible transitions between the current symbol and the next symbol. The *differential pulse code modulation* (DPCM) predictive coder makes use of the Markov source model. DPCM is used in the MPEG-1 and H.261 standards to encode the set of quantized DC coefficients generated by the discrete cosine transforms.

The DPCM predictive encoder modifies the use of the Markov source model considerably in order to reduce its complexity. It does not rely on the actual Markov source statistics at all, and simply creates a linear weighting of the last m symbols (m-th order) to predict the next state. This significantly reduces the complexity of using Markov source prediction at the expense of an increase in the bit rate. DPCM encodes the *differential signal* d between the actual value and the predicted value, i.e., $d = e - \hat{e}$, where the prediction $\hat{e}$ is a linear weighting of m previous values. The resulting differential signal d generally has reduced entropy as compared to the original source. DPCM is used in conjunction with a VLC encoder to reduce the bit rate. The simplicity and entropy reduction capability of DPCM makes it a good choice for use in real-time compression systems. Third order predictors ($m=3$) have been shown to provide good performance on natural images [10].

5.2 Block Transform Coding — The Discrete Cosine Transform

Block transform coding is widely used in image and video compression systems. The transforms used in video encoders are *unitary*, which means that the transform operation has an inverse operation that uniquely reconstructs the original input. The *discrete cosine transform* (DCT), successively operates on 8×8 image blocks, and is used in the H.261, H.263, and MPEG standards. Block transforms make use of the high degree of correlation between adjacent image pixels to provide *energy compaction* or coding gain in the transformed domain. The *block transform coding gain* G_{TC} is defined as the logarithmic ratio of the arithmetic and geometric means of the transformed block variances, i.e.,

$$G_{TC} = 10 \log_{10} \left[\frac{\frac{1}{N} \sum_{i=0}^{N-1} \sigma_i^2}{\left(\prod_{i=0}^{N-1} \sigma_i^2 \right)^{1/N}} \right], \tag{6}$$

where the transformed image block is divided into N subbands, and σ_i^2 is the variance of each subband i, for $0 \le i \le N-1$. G_{TC} also measures the gain of block transform coding over PCM coding. The coding gain generated by a block transform is realized by packing most the original signal energy content into a small number of transform coefficients. This results in a lossless representation of the original signal that is more suitable for quantization. That is, there may be many transform coefficients containing little or no energy that can be completely eliminated. Spatial transforms should also be orthonormal, i.e., generate uncorrelated coefficients, so that simple scalar quantization can be used to quantize the coefficients independently.

The *Karhunen-Loève transform* (KLT) creates uncorrelated coefficients, and is optimal in the energy packing sense. But the KLT is not widely used in practice. It requires the calculation of the image block covariance matrix so that its unitary orthonormal eigenvector matrix can be used to generate the KLT coefficients. This calculation (for which no fast algorithms exist), and the transmission of the eigenvector matrix is required for every transformed image block.

The DCT is the most widely used block transform for digital image and video encoding. It is an orthonormal transform, and has been found to perform close to the KLT [11] for first-order Markov sources. The DCT is defined on an 8×8 array of pixels,

$$F(u,v) = \frac{1}{4} C_u C_v \sum_{i=0}^{7} \sum_{j=0}^{7} f(i,j) \cos\left(\frac{(2i+1)u\pi}{16} \right) \cos\left(\frac{(2j+1)v\pi}{16} \right) \tag{7}$$

and the inverse IDCT is defined as,

$$f(i,j) = C_u C_v \sum_{u=0}^{7} \sum_{v=0}^{7} F(u,v) \cos\left(\frac{(2i+1)u\pi}{16} \right) \cos\left(\frac{(2j+1)v\pi}{16} \right) \tag{8}$$

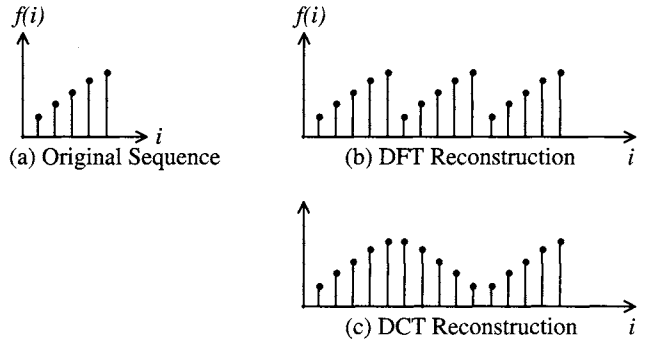


FIGURE 3 Reconstruction periodicity of DFT vs DCT.

with

$$C_u = \frac{1}{\sqrt{2}} \quad \text{for } u = 0, \quad C_u = 1 \quad \text{otherwise}$$

$$C_v = \frac{1}{\sqrt{2}} \quad \text{for } v = 0, \quad C_v = 1 \quad \text{otherwise}$$

where i and j are the horizontal and vertical indices of the 8×8 spatial array, and u and v are the horizontal and vertical indices of the 8×8 coefficient array. The DCT is the chosen method for image transforms for a couple of important reasons. The DCT has fast O($n$log$n$) implementations using real calculations. It is even simpler to compute than the DFT because it does not require the use of complex numbers.

The second reason for its success is that the reconstructed input of the *inverse DCT* (IDCT) tends not to produce any significant discontinuities at the block edges. Finite discrete transforms create a reconstructed signal that is periodic. Periodicity in the reconstructed signal can produce discontinuities at the periodic edges of the signal or pixel block. The DCT is not as susceptible to this behavior as the *discrete Fourier transform* (DFT). Since the cosine function is real and even, i.e., $\cos(x) = \cos(-x)$, and the input $F(u, v)$ is real, the IDCT generates a function that is even and periodic in $2n$, where n is the length of the original sequence. On the other hand, the IDFT produces a reconstruction that is periodic in n, but and necessarily even. This phenomenon is illustrated in Fig. 3 for the one dimensional signal $f(i)$.

The original finite sequence $f(i)$ depicted in part (a) is transformed and reconstructed in (b) using the DFT-IDFT transform pairs, and in (c) using the DCT-IDCT transform pairs. The periodicity of the IDFT in (b) is five samples long, and illustrates the discontinuity introduced by the discrete transform. The periodicity of the IDCT in (c) is 10 samples long, due to the evenness of the DCT operation. Discontinuities introduced by the DCT are generally less severe than the DFT. The importance of this property of the DCT is that reconstruction errors and blocking artifacts are less severe in comparison to the DFT. Blocking artifacts are visually striking and occur due to the loss of high frequency components that are either quantized or eliminated from the DCT coefficient array. The DCT minimizes blocking artifacts as compared to the DFT because it doesn't introduce the same level of reconstruction discontinuities at the block edges. Figure 4 depicts blocking artifacts introduced by gross quantization of the DCT coefficients.

This section ends with an example of the energy packing capability of the DCT. Figure 5 depicts the DCT transform operation. The original 8×8 image sub-block from the Lena image is displayed in part (a), and the DCT transformed coefficient array is displayed in part (b).

The original image sub-block in (a) contains large values in every position is not very suitable for spatial compression in this format. The coefficient matrix (b) concentrates most of the signal energy in the top left quadrant. The signal frequency coordinates $(u, v) = (0, 0)$ start at the upper left position. The DC component equals 1255 and contains the vast majority of the signal energy by itself. This dynamic range and concentration of energy should yield a significant reduction in non-zero values and bit rate after the coefficients are quantized.

(a) Original (b) Reconstructed

FIGURE 4 Severe blocking artifacts introduced by gross quantization of DCT coefficients.

$$f(i,j) = \begin{bmatrix} 136 & 141 & 143 & 153 & 152 & 154 & 154 & 156 \\ 143 & 150 & 153 & 156 & 160 & 156 & 155 & 155 \\ 149 & 155 & 161 & 163 & 158 & 155 & 156 & 155 \\ 158 & 161 & 162 & 161 & 160 & 158 & 160 & 157 \\ 157 & 161 & 160 & 162 & 161 & 157 & 154 & 155 \\ 160 & 160 & 161 & 160 & 160 & 156 & 156 & 156 \\ 160 & 161 & 160 & 161 & 161 & 157 & 157 & 156 \\ 162 & 162 & 161 & 161 & 162 & 157 & 157 & 157 \end{bmatrix}$$

(a) Original Lena 8×8 image sub-block

$$F(u,v) = \begin{bmatrix} 1255 & -8 & -9 & -6 & 1 & -1 & -3 & 1 \\ -26 & -20 & -5 & 4 & -1 & 1 & 0 & 1 \\ -9 & -5 & 1 & -1 & 0 & 0 & -1 & 0 \\ -6 & -2 & 0 & 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 2 & 0 & -1 & -1 & 0 \\ -2 & 1 & 2 & 0 & 1 & 1 & 0 & -1 \\ -1 & 0 & 0 & -2 & 0 & 0 & 1 & -1 \\ 1 & 0 & -1 & -2 & 0 & 1 & -1 & 0 \end{bmatrix}$$

(b) DCT Coefficients

FIGURE 5 8 × 8 discrete cosine transform.

5.3 Quantization

The quantization stage of the video encoder creates a lossy representation of the input. The input as discussed earlier should be conditioned with a particular method of quantization in mind. And vice versa, the quantizer should be well matched to the characteristics of the input in order to meet or exceed the rate-distortion performance requirements. As always is the case, the quantizer has an effect on system performance that must be taken under consideration. Simple scalar versus vector quantization implementations can have significant system performance implications.

Scalar and Vector represent the two major types of quantizers. These can be further classified as memoryless or containing memory, and symmetric or nonsymmetric. Scalar quantizers control the values taken by a single variable. The quantizer defined by the MPEG-1 encoder scales the DCT transform coefficients. Vector quantizers operate on multiple variables, i.e., a vector of variables, and become very complex as the number of variables increases. This discussion will introduce the reader to the basic scalar and vector quantizer concepts that are relevant to image and video encoding.

The uniform scalar quantizer is the most fundamental scalar quantizer. It possesses a nonlinear staircase input-output characteristic that divides the input range into output levels of equal size. In order for the quantizer to effectively reduce the bit rate, the number of output values should be much less that the number of input values. The reconstruction values are chosen to be at the midpoint of the output levels. This choice is expected to minimize the reconstruction MSE when the quantization errors are uniformly distributed. The quantizers specified in the H.261, H.263, MPEG-1, and MPEG-2 video coders are *nearly* uniform. They have constant step sizes except for the larger *dead zone* area (the input range for which the output is zero).

Non-uniform quantization is typically used for non-uniform input distributions, such as natural image sources. The scalar quantizer that produces the minimum MSE for

a non-uniform input distribution will have non-uniform steps. As compared to the uniform quantizer, the non-uniform quantizer has increasingly better MSE performance as the number of quantization steps increases. The Lloyd-Max [12] is a scalar quantizer design that utilizes the input distribution to minimize the MSE for a given number of output levels. The Lloyd-Max places the reconstruction levels at the centroids of the adjacent input quantization steps. This minimizes the total absolute error within each quantization step based upon the input distribution.

Vector quantizers (discussed in Section 5.3) decompose the input into a length n vector. An image for instance, can be divided into $M \times N$ blocks of n pixels each, or the image block can be transformed into a block of transform coefficients. The resulting vector is created by scanning the two dimensional block elements into a vector of length n. A vector $\mathbf{X}$ is quantized by choosing a codebook vector representation $\hat{\mathbf{X}}$ that is its "closest match." The closest match selection can be made by minimizing an error measure, i.e., choose $\hat{\mathbf{X}} = \hat{\mathbf{X}}_i$ such that the MSE over all codebook vectors is minimized,

$$\hat{\mathbf{X}} = \hat{\mathbf{X}}_i : \min_i \mathrm{MSE}(\mathbf{X}, \hat{\mathbf{X}}_i) = \min_i \frac{1}{n} \sum_{j=1}^{n} (x_j - \hat{x}_j)^2. \quad (9)$$

The index i of the vector $\hat{\mathbf{X}}_i$ denotes the codebook entry that is used by the receiver to decode the vector. Obviously the complexity of the decoder is much simpler than the encoder. The size of the codebook dictates both the coding efficiency and reconstruction quality. The raw bit rate of a vector quantizer is,

$$\mathrm{bitrate}_{\mathrm{VQ}} = \frac{\log_2 m}{n} \text{ bits/pixel}, \quad (10)$$

where $\log_2 m$ is the number of bits required to transmit the index i of the codebook vector $\hat{\mathbf{X}}_i$. The codebook construction includes two important issues that are pertinent to the performance of the video coder. The set of vectors that are included in the codebook determine the bit rate and distortion characteristics of the reconstructed image sequence. The codebook size and structure determines the search complexity to find the minimum error solution for Eq. (9). Two important VQ codebook designs are the *Linde-Buzo-Gray* (LBG) [13] and *tree search VQ* (TSVQ) [14]. The LBG design is based on the Lloyd-Max scalar quantizer algorithm. It is widely used because the system parameters can be generated via the use of an input "training set" instead of the true source statistics. The TSVQ design reduces VQ codebook search time by using m-ary tree structures and searching techniques.

5.4 Motion Compensation and Estimation

Motion compensation [15] is a technique created in the 1960s which is used to increase the efficiency of video encoders. Motion compensated video encoders are implemented in three stages. The first stage estimates objective motion (motion estimation) between the previously reconstructed frame and the current frame. The second stage creates the current frame prediction (motion compensation) using the motion estimates and the previously reconstructed frame. The final stage differentially encodes the prediction and the actual current frame as the prediction error. Therefore, the receiver reconstructs the current image only using the VLC encoded motion estimates and the spatially and VLC encoded prediction error.

Motion estimation and compensation are common techniques used to encode the temporal aspect of a video signal. As discussed earlier, block based motion compensation and motion estimation techniques used in video compression systems are capable of the largest reduction in the raw signal bit rate. Typical implementations generally out-perform pure spatial encodings by a factor of three or more. The interframe redundancy contained in the temporal dimension of a digital image sequence accounts for the impressive signal compression capability that can be achieved by video encoders. Interframe redundancy can be simply modeled as static backgrounds and moving foregrounds to illustrate the potential temporal compression that can be realized. Over a short period of time, image sequences can be described as a static background with moving objects in the foreground. If the background does not change between two frames, their difference is zero, and the two background frames can essentially be encoded as one. Therefore the compression ratio increase is proportional to two times the spatial compression achieved in the first frame. In general, unchanging or static backgrounds can realize additive coding gains, i.e.,

$$\text{Static Back ground Coding Gain} \propto N \bullet (\text{Spatial Compression Ratio of Background Frame}) \quad (11)$$

where N is the number of static background frames being encoded. Static backgrounds occupy a great deal of the image area, and are typical of both natural and animated image sequences. Some variation in the background always occurs due to random and systematic fluctuations. This tends to reduce the achievable background coding gain.

Moving foregrounds are modeled as non-rotational rigid objects that move independent of the background. Moving objects can be detected by matching the foreground object between two frames. A perfect match results in zero difference between the two frames. In theory, moving foregrounds

can also achieve additive coding gain. In practice, moving objects are subject to occlusion, rotational and non-rigid motion, and illumination variations that reduce the achievable coding gain. Motion compensation systems that make use of motion estimation methods leverage both background and foreground coding gain. They provide pure interframe differential encoding when two backgrounds are static, i.e., the computed motion vector is (0, 0). And the motion estimate computed in the case of moving foregrounds generates the minimum distortion prediction.

Motion estimation is an interframe prediction process falling in two general categories; pel-recursive algorithms [16] and block-matching algorithms (BMA) [17]. The pel-recursive methods are very complex and inaccurate, which restrict their use in video encoders. Natural digital image sequences generally display ambiguous object motions that adversely affect the convergence properties of pel-recursive algorithms. This has led to the introduction of *block-matching motion estimation* which is tailored for encoding image sequences. Block-matching motion estimation assumes that the objective motion being predicted is rigid and non-rotational. The block size of the BMA for the MPEG, H.261, and H.263 encoders is defined as 16×16 luminance pixels. MPEG-2 also supports 16×8 pixel blocks.

BMAs predict the motion of a block of pixels between two frames in an image sequence. The prediction generates a pixel displacement or motion vector whose size is constrained by the search neighborhood. The search neighborhood determines the complexity of the algorithm. The search for the best prediction ends when the best block match is determined within the search neighborhood. The best match can be chosen as the minimum MSE, which for a full search is computed for each block in the search neighborhood, i.e.,

$$\text{Best Match}_{\text{MSE}} = \min_{m,n} \frac{1}{N^2} \sum_{i=1}^{M} \sum_{j=1}^{N} [I^k(i,j) - I^{k-l}(i+m, j+n)]^2, \tag{12}$$

where k is the frame index, l is the temporal displacement in frames, M is the number of pixels in the horizontal direction and N is the number of pixels in the vertical direction of the image block, i and j are the pixel indices within the image block, and m and n are the indices of the search neighborhood in the horizontal and vertical directions. Therefore the best match motion vector estimate $MV(m=x, n=y)$ is the pixel displacement between the block $I^k(i,j)$ in frame k, and the best matched block $I^{k-l}(i+x, j+y)$ in the displaced frame $k-l$. The best match is depicted in Fig. 6.

In cases where the block motion is not uniform or if the scene changes, the motion estimate may in fact increase the bit rate over the corresponding spatial encoding of the block. In the case where the motion estimate is not effective, the video encoder does not use the motion estimate, and encodes the block using the spatial encoder.

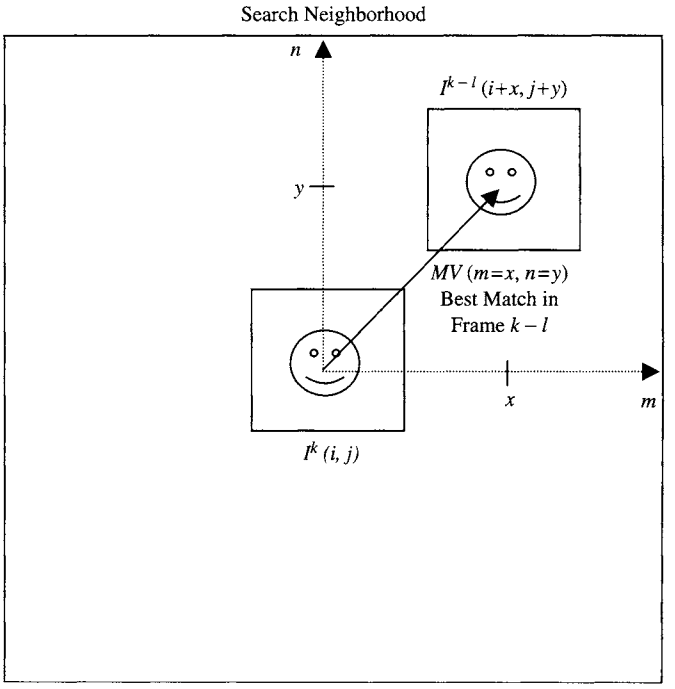


FIGURE 6 Best match motion estimate.

The search space size determines the complexity of the motion estimation algorithm. Full search methods are costly and are not generally implemented in real-time video encoders. Fast searching techniques can considerably reduce computational complexity while maintaining good accuracy. These algorithms reduce the search process to a few sequential steps in which each subsequent search direction is based on the results of the current step. The procedures are designed to find local optimal solutions and cannot guarantee selection of the global optimal solution within the search neighborhood. The *logarithmic* search [18] algorithm proceeds in the direction of minimum distortion until the final optimal value is found. Logarithmic searching has been implemented in some MPEG encoders. The *three-step* search [19] is a very simple technique that proceeds along a best match path in three steps in which the search neighborhood is reduced for each successive step. Figure 7 depicts the three-step search algorithm.

A 14×14 pixel search neighborhood is depicted. The search area sizes for each step are chosen so that the total search neighborhood can be covered in finding the local minimum. The search areas are square. The length of the sides of the search area for step 1 are chosen to be larger than or equal to ½ the length of the range of the search neighborhood (in this example the search area is 8×8). The length of the sides are successively reduced by ½ after each of the first two steps are completed. Nine points for each step are compared using the

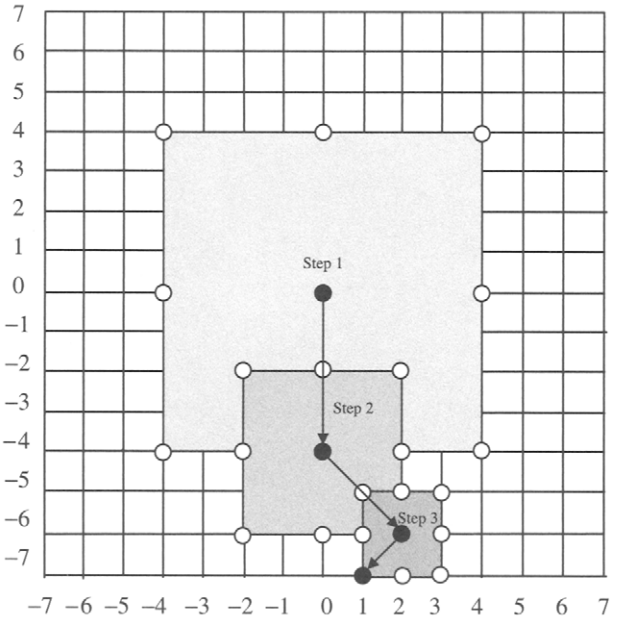


FIGURE 7 Three-step search algorithm pictorial.

matching criteria. These consist of the central point and eight equally spaced points along the perimeter of the search area. The search area for step 1 is centered on the search neighborhood. The search proceeds in by centering the search area for the next step over the best match from the previous step. The overall best match is the pixel displacement chosen to minimize the matching criteria in step 3. The total number of required comparisons for the three-step algorithm is 25. That represents an 87% reduction in complexity versus the full search method for a 14×14 pixel search neighborhood.

6 Video Encoding Standards and H.261

The major internationally recognized video compression standards have been developed by the *International Standardization Organization* (ISO), the *International Electrotechnical Commission* (IEC), and the *International Telecommunications Union* (ITU) standards organizations. The *Moving Pictures Experts Group* (MPEG) is a working group operating within ISO and IEC. Since starting its activity in 1988, MPEG has produced ISO/IEC 11172 (MPEG-1, 1992), ISO/IEC 13818 (MPEG-2, 1994), ISO/IEC 14496 (MPEG-4, 1999), and ISO/IEC 15938 (MPEG-7, 2001).

The MPEG-1 specification was motivated by T1 transmission speeds, the CD-ROM, and the multimedia capabilities of the desktop computer. It is intended for video coding up to the rate of 1.5 Mbps, and is composed of five sections: System Configurations, Video Coding, Audio Coding, Compliance Testing, and Software for MPEG-1 Coding. The standard does not specify the actual video coding process, but only the syntax and semantics of the bit stream, and the video generation at the receiver. It does not accommodate interlaced video, and only supports CIF quality format at 25 or 30 fps.

Activity for MPEG-2 was started in 1991. It was targeted for higher bit rates, broadcast video, and a variety of consumer and telecommunications video and audio applications. The syntax and technical contents of the standard were frozen in 1993. It is composed of four parts: Systems, Video, Audio, and Conformance Testing. MPEG-2 was also recommended by the ITU as H.262.

The MPEG-4 project is targeted to enable content developers and users to achieve various forms of interactivity with the audio-visual content of a scene, and to mix synthetic and natural audio and video information in a seamless way. MPEG-4 technology comprises two major parts; a set of coding tools for audiovisual objects, and a syntactic language to describe both the coding tools and the coded objects. From a technical viewpoint, the most notable departure from traditional coding standards is the ability for a receiver to download the description of the syntax used to represent the audio-visual information. The visual information is not restricted to have the format of conventional video, i.e., it may not necessarily be frame-based, but can incorporate audio and/or video foreground and background objects, which can produce significant improvements in both encoder efficiency and functionality.

MPEG-7 is formally named "Multimedia Content Description Interface," and is a common way of describing multimedia content data that is used to access and interpret content by a computer program. Since much of the value of multimedia content can be derived from its accessibility, MPEG-7 strives to define common data access methods to maximize the value of multimedia information regardless of the technologies encompassed by the source and destination, or the specific applications using its services. In order to meet these requirements MPEG-7 has created a hierarchic framework that can handle many levels of description. In addition other types of descriptive data are defined, such as, coding formats, data access conditions, parental ratings, relevant links, and the overall context. MPEG-7 is made up of three main elements that include description tools, storage and transmission system tools, and a language to define the MPEG-7 syntax. These elements provide the flexibility to meet the stated requirements.

Work on the latest MPEG standard began in 2000. It was started as an extension to MPEG-7 and is known as MPEG-21 (ISO/IEC 21000). It is focused on defining the common content and user access model addressing the vast proliferation of new and old multimedia distribution and reception technologies. MPEG-21 specifically looks to define the technology needed to support users to exchange, access, consume, trade, and otherwise manipulate Digital items in an efficient, transparent and interoperable way. Digital items

are defined to be the fundamental unit of distribution and transaction, i.e., content (Web page, picture, movie, etc.).

The precursor to the MPEG video encoding standards development is the H.261 encoder which contains many video of the coding methods and techniques later adopted by MPEG. The ITU Recommendation H.261 was adopted in 1990, and specifies a video encoding standard for videoconferencing and videophone services for transmission over *integrated services digital network* (ISDN) at $p \times 64$ kbps, $p = 1, \ldots, 30$. H.261 describes the video compression methods that were later adopted by the MPEG standards and is presented in the following section. The ITU Experts Group for Very Low Bit-Rate Video Telephony (LBC) has produced the H.263 Recommendation for *Public Switched Telephone Networks* (PSTN), which was finalized in December 1995 [19]. It is and extended version of H.261 supporting bi-directional motion compensation and sub-QCIF formats. The encoder is based on hybrid DPCM/DCT coding and improvements targeted to generate bit rates of less than 64 Kbps.

6.1 The H.261 Video Encoder

The H.261 recommendation [4] is targeted at the videophone and videoconferencing application market running on connection based ISDN at $p \times 64$ kbps, $p = 1, \ldots, 30$. It explicitly defines the encoded bit stream syntax and *decoder*, while leaving the encoder design to be compatible with the decoder specification. The video encoder is required to carry a delay of less than 150 msec so that it can operate in real-time bi-directional videoconferencing applications. H.261 is part of a group of related ITU recommendations that define visual telephony systems. This group includes:

1. H.221 — Defines the frame structure for an audiovisual channel supporting 64–1920 Kbps.
2. H.230 — Defines frame control signals for audiovisual systems.
3. H.242 — Defines audiovisual communications protocol for channels supporting up to 2 Mbps.
4. H.261 — Defines the video encoder/decoder for audiovisual services at $p \times 64$ Kbps.
5. H.320 — Defines narrow-band audiovisual terminal equipment for $p \times 64$ Kbps transmission.

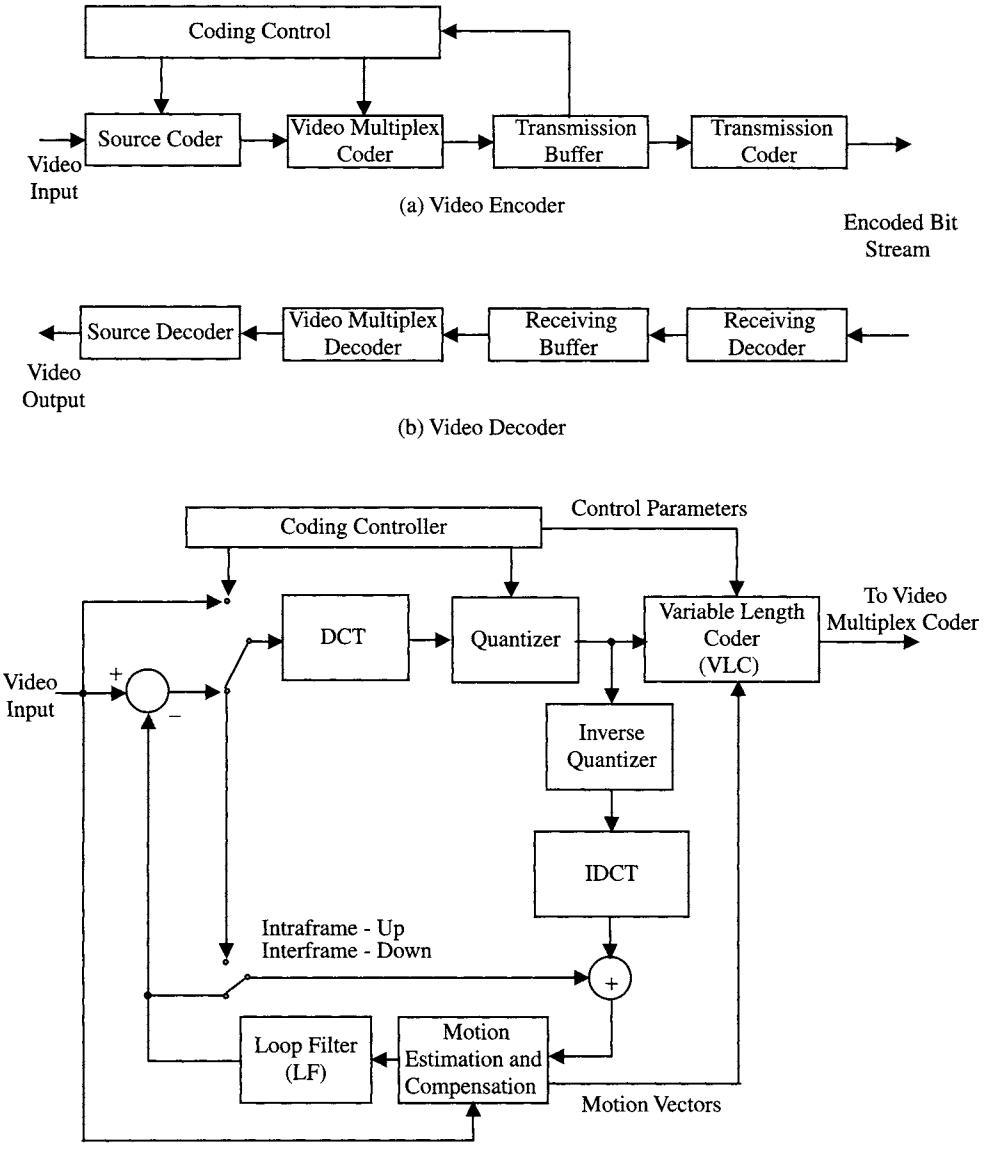
The H.261 encoder block diagrams are depicted in Fig. 8 (a) and (b). An H.261 source coder implementation is depicted in (c). The source coder implements the video encoding algorithm that includes the spatial encoder, the quantizer, the temporal prediction encoder, and the VLC. The spatial encoder is defined to use the two dimensional 8×8 pixel block DCT and a nearly uniform scalar quantizer using a possible 31 step sizes to scale the AC and interframe DC coefficients. The resulting quantized coefficient matrix is zigzag scanned into a vector that is VLC coded using a hybrid modified run length and Huffman coder. Motion compensation is optional. Motion estimation is only defined in the forward direction because H.261 is limited to real-time videophone and videoconferencing. The recommendation does not specify the motion estimation algorithm or the conditions for the use of intraframe versus interframe encoding.

The video multiplex coder creates a H.261 bit stream that is based on the data hierarchy described below. The transmission buffer is chosen not to exceed the maximum coding delay of 150 msec, and is used to regulate the transmission bit rate via the coding controller. The transmission coder embeds an error correction code (ECC) into the video bit stream that provides error resilience, error concealment, and video synchronization.

H.261 supports most of the internationally accepted digital video formats. These include, CCIR 601, SIF, CIF, and QCIF. These formats are defined for both NTSC and PAL broadcast signals. The CIF and QCIF formats were adopted in 1984 by H.261 in order to support 525-line NTSC and 625-line PAL/SECAM video formats. The CIF and QCIF operating parameters can be found in Table 4. The raw data rate for 30 fps CIF is 37.3 Mbps and 9.35 Mbps for QCIF. CIF is defined for use in channels in which $p \geq 6$ so that the required compression ratio for 30 fps is less than 98:1. CIF and QCIF formats support frame rates of 30, 15, 10, and 7.5 fps, which allows the H.261 encoder to achieve greater coding efficiency by skipping the encoding and transmission of whole frames. H.261 allows 0, 1, 2, 3 or more frames to be skipped between transmitted frames.

H.261 specifies a set of encoder protocols and decoder operations that every compatible system must follow. The H.261 *video multiplex* defines the data structure hierarchy that the decoder can interpret unambiguously. The video data hierarchy defined in H.261 is depicted in Fig. 9. They are the picture layer, group of block (GOB) layer, macroblock (MB) layer, and the basic (8×8) block layer. Each layer is built from the previous or lower layers, and contains its associated data payload, and header that describes the parameters used to generate the bit stream. The basic 8×8 block is used in intraframe DCT encoding. The macroblock is the smallest unit for selecting intraframe or interframe encoding modes. It is made up of four adjacent 8×8 luminance blocks and two subsampled 8×8 color difference blocks (C_B and C_R as defined in Table 4) corresponding to the luminance blocks. The GOB is made up of 176×48 pixels (33 macroblocks) and is used to construct the 352×288 pixel CIF or 176×144 pixel QCIF picture layer.

The headers for the GOB and picture layers contain start codes so that the decoder can re-synchronize when errors occur. They also contain other relevant information required to reconstruct the image sequence. The following parameters used in the headers of the data hierarchy complete the H.261 video multiplex.



(a) Video Encoder

(b) Video Decoder

(c) H.261 Source Encoder Implementation

FIGURE 8 ITU-T H.261 block diagrams.

Picture Layer

- Picture start code (PSC), 20-bit synchronization pattern (0000 0000 0000 0001 0000).
- Temporal reference (TR), 5-bit input frame number.
- Type information (PTYPE), indicates source format, CIF=1 QCIF=0, and other controls.
- User-inserted bits.

GOB Layer

- Group of blocks start code (GBSC), 16-bit synchronization code (0000 0000 0000 0001).
- Group number (GN), 4-bit address representing the 12 GOBs per CIF frame.
- Quantizer information (GQUANT), indicates one of 31 quantizer step sizes to be used in a GOB unless overridden by macroblock MQUANT parameter.
- User-inserted bits.

Macroblock Layer

- Macroblock address (MBA), is the position of a macroblock within a GOB.
- Type information (MTYPE), for one of 10 encoding modes used for the macroblock. This includes permutations of intraframe, interframe, motion compensation (MC), and loop filtering (LF). A pre-specified VLC is used to encode these modes.

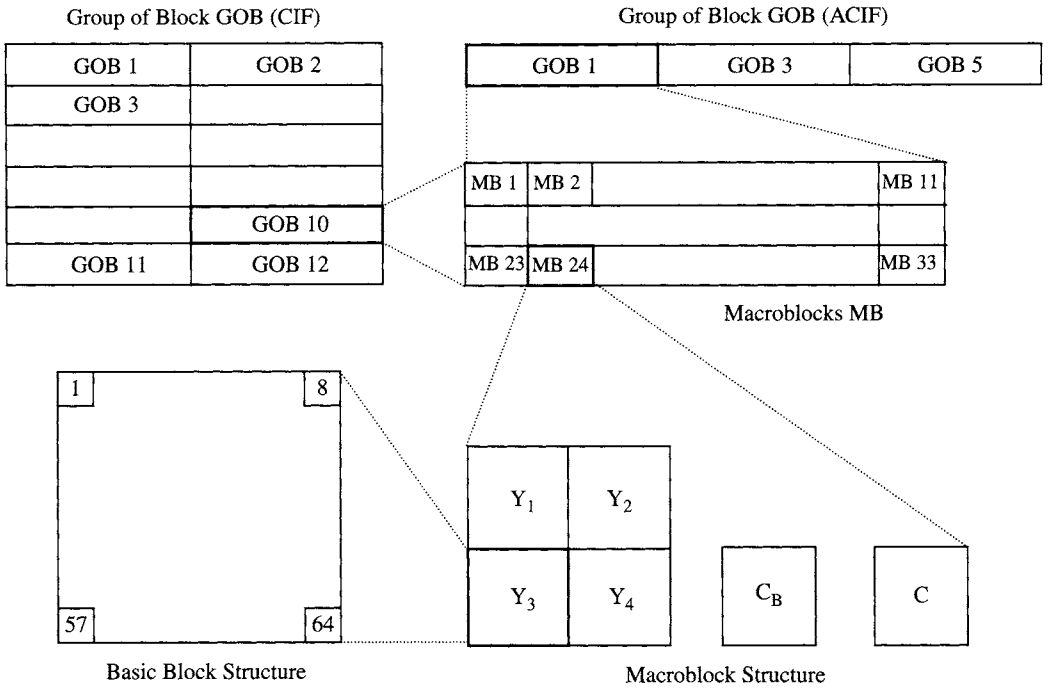


FIGURE 9 H.261 block hierarchy.

- Quantizer (MQUANT), 5-bit normalized quantizer step size from 1–31.
- Motion vector data (MVD), up to 11-bit VLC describing the differential displacement.
- Coded block pattern (CBP), up to 9-bit VLC indicating the location of the encoded blocks in the macroblock.

Block Layer

- Transform coefficients (TCOEFF), are zigzag scanned and can be 8-bits fixed or up to 13-bit VLC.
- End of block (EOB), symbol.

The H.261 bit stream also specifies transmission synchronization and error code correction using a BCH code [20] that is capable of correcting 2-bit errors in every 511-bit block. It inserts 18 parity bits for every 493 data bits. A synchronization bit is added to every codeword to be able to detect the BCH codeword boundaries. The transmission synchronization and encoding also operates on the audio and control information specified by the ITU H.320 Recommendation.

The H.261 video compression algorithm depicted in Fig. 7 (c) is specified to operate in intraframe and interframe encoding modes. The intraframe mode provides spatial encoding of the 8×8 block, and uses the two-dimensional DCT. Interframe mode encodes the prediction error, with motion compensation being optional. The prediction error is optionally DCT encoded. Both modes provide options that affect the performance and video quality of the system. The motion estimate method, mode selection criteria, and block transmission criteria are not specified although the ITU has published reference models [21, 22] that make particular implementation recommendations. The coding algorithm used in the ITU-T *Reference Model 8* (RM8) [22] is summarized in three steps, and is followed by an explanation of its important encoding elements.

1. The motion estimator creates a displacement vector for each macroblock. The motion estimator generally operates on the 16×16 pixel luminance macroblock. The displacement vector is an integer value between ± 15, which is the maximum size of the search neighborhood. The motion estimate is scaled by a factor of 2 and applied to the C_R and C_B component macroblocks.
2. The compression mode for each macroblock is selected using a minimum error criteria that is based upon the *displaced macroblock difference* (DMD),

$$DMD(i,j,k) = b(i,j,k) - b(i - d_1, j - d_2, k - 1) \qquad (13)$$

where b is a 16×16 macroblock, i and j are its spatial pixel indices, k is the frame index, and d_1 and d_2 are the pixel displacements of the macroblock in the previous frame. The displacements range from $-15 \leq d_1, d_2 \leq +15$. When d_1 and d_2 are set to zero, the DMD becomes the *macroblock difference* (MD). The compression mode determines the operational encoder elements that are used for the current frame. The H.261 compression modes are depicted in Table 5.

TABLE 5 H.261 Macroblock video compression modes

Mode	MQUANT	MVD	CBP	TCOEFF
Intra				√
Intra	√			√
Inter			√	√
Inter	√		√	√
Inter + MC		√		
Inter + MC		√	√	√
Inter + MC	√	√	√	√
Inter + MC + LF		√		
Inter + MC + LF		√	√	√
Inter + MC + LF	√	√	√	√

3. The *video multiplex coder* processes each macroblock to generate the H.261 video bitstream whose elements are discussed above.

There are five basic MTYPE encoding mode decisions that are carried out in step 2. These are,

- Use intraframe or interframe mode?
- Use motion compensation?
- Use a coded block pattern (CBP)?
- Use loop-filtering?
- Change quantization step size MQUANT?

To select the macroblock compression mode, the *variances* (VAR) of the input macroblock, the macroblock Difference (MD) and the displaced macroblock difference (DMD) (as determined by the best motion estimate) are compared as follows,

1. If VAR(DBD) < VAR(MD) then interframe + motion compensation (Inter + MC) coding is selected. In this case, the motion vector data (MVD) is transmitted. Table 5 indicates that there are three Inter + MC modes that allow for the transmission of the prediction error (DMD) with or without DCT encoding of some or all of the four 8×8 basic blocks.
2. "VAR input" is defined as the variance of the input macroblock. If VAR input < VAR(DMD) and VAR input < VAR(MD) then the intraframe mode (Intra) is selected. Intraframe mode uses DCT encoding of all four 8×8 basic blocks.
3. If VAR(MD) < VAR(DMD) then interframe mode (Inter) is selected. This mode indicates that the motion vector is zero, and that some or all of the 8×8 prediction error (MD) blocks can be DCT encoded.

The transform coefficient coded block pattern (CBP) parameter is used to indicate whether a basic block is reconstructed using the corresponding basic block from the previous frame, or if it is encoded and transmitted. In other words, no basic block encoding is used when the block content does not change significantly. The CPB parameter encodes 63 combinations of the four luminance blocks and two color difference blocks using a variable length code. The conditions for using CBP are not specified in the H.261 recommendation.

Motion compensated blocks can be chosen to be low pass filtered before the prediction error is generated by the feedback loop. This mode is denoted as Inter + MC + LF in Table 5. The low pass filter is intended to reduce the quantization noise in the feedback loop, as well as the high frequency noise and artifacts introduced by the motion compensator. H.261 defines loop filtering as optional and recommends a separable two-dimensional spatial filter design which is implemented by cascading two identical one dimensional *finite impulse response* (FIR) filters. The coefficients of the 1D filter are [1, 2, 1] for pixels inside the block, and [0, 1 , 0] (no filtering) for pixels on the block boundary.

The MQUANT parameter is controlled by the state of the transmission buffer in order to prevent overflow or underflow conditions. The dynamic range of the DCT macroblock coefficients extends between $[-2047, \ldots, 2047]$. They are quantized to the range $[-127, \ldots, 127]$ using one of the 31 quantizer step sizes as determined by the GQUANT parameter. The step size is an even integer in the range of $[2, \ldots, 62]$. GQUANT can be overridden at the Macroblock layer by MQUANT to clip or expand the range prescribed by GQUANT so that the transmission buffer is better utilized. The ITU-T RM8 *liquid level control model* specifies the inspection of 64 Kbit transmission buffers after encoding 11 macroblocks. The step size of the quantizer should be increased (decreasing the bitrate) if the buffer is full, and vice versa, the step size should be decreased (increasing the bitrate) if the buffer is empty. The actual design of the rate control algorithm is not specified.

The DCT macroblock coefficients are subjected to variable thresholding before quantization. The threshold is designed to increase the number of zero valued coefficients, which in turn increases the number of the zero run-lengths and VLC coding efficiency. The ITU-T *Reference Model 8* provides an example thresholding algorithm for the H.261 encoder. Nearly uniform scalar quantization using a dead zone is applied after the thresholding process. All the coefficients in the luminance and chrominance Macroblocks are subjected to the same quantizer except for the intraframe DC coefficient. The intraframe DC coefficient is quantized using a uniform scalar quantizer whose step size is 8. The quantizer decision levels are not specified, but the reconstruction levels are defined in H.261 as follows,
For case QUANT odd

$$\text{REC\_LEVEL} = \text{QUANT} \times (2 \times \text{COEFF\_VALUE} + 1),$$
$$\text{for COEFF\_LEVEL} > 0,$$

$$\text{REC\_LEVEL} = \text{QUANT} \times (2 \times \text{COEFF\_VALUE} - 1),$$

$$\text{for COEFF\_LEVEL} < 0.$$

For case QUANT even

$$\text{REC\_LEVEL} = \text{QUANT} \times (2 \times \text{COEFF\_VALUE} + 1) - 1,$$

$$\text{for COEFF\_LEVEL} > 0,$$

$$\text{REC\_LEVEL} = \text{QUANT} \times (2 \times \text{COEFF\_VALUE} - 1) + 1,$$

$$\text{for COEFF\_LEVEL} < 0.$$

If COEFF_VALUE = 0, then REC_LEVEL = 0, where REC_LEVEL is the reconstruction value, QUANT is ½ the macroblock quantization step size ranging from 1–31, and COEFF_VALUE is the quantized DCT coefficient.

To increase the coding efficiency, lossless variable length coding is applied to the quantized DCT coefficients. The coefficient matrix is scanned in a zig-zag manner in order to maximize the number of zero coefficient run-lengths. The VLC encodes events defined as the combination of a run-length of zero coefficients preceding a nonzero coefficient, and the value of the nonzero coefficient, i.e., EVENT = (RUN, VALUE). The VLC EVENT tables are defined in [4].

7 Closing Remarks

Digital video compression, although only recently becoming an internationally standardized technology, is strongly based upon the information coding technologies researched over the last 40 years. The large variety of bandwidth and video quality requirements for the transmission and storage of digital video information has demanded that a variety of video compression techniques and standards be developed. The major international standards recommended by ISO and the ITU make use of common video coding methods. The generalized digital video encoder introduced in Section 2, illustrates the spatial and temporal video compression elements that are central to the current MPEG-1, MPEG-2/H.262, MPEG-4, H.261, and H.263 standards that have been developed over the past decade. They address a vast landscape of application requirements, from low to high bitrate environments, as well as stored video and multimedia to real-time videoconferencing and high quality broadcast television.

References

[1] ISO/IEC 11172 Information Technology: coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s, 1993.

[2] ISO/IEC JTC1/SC29/WG11, CD 13818: Generic coding of moving pictures and associated audio, 1993.

[3] ISO/IEC JTC1/SC29/WG11, CD 14496: Coding of audio-visual objects, 1999.

[4] CCITT Recommendation H.261: "Video Codec for Audio Visual Services at $p \times 64$ kbits/s," COM XV-R 37-E, 1990.

[5] H. Hseuh-Ming and J.W. Woods, "*Handbook of Visual Communications*," Chapter 6, Academic Press Inc., San Diego, CA., 1995.

[6] J.W. Woods, "*Subband Image Coding*," Kluwer Academic Publishers, Norwell, MA., 1991.

[7] L. Wang and M. Goldberg, "Progressive image transmission using vector quantization on images in pyramid form," *IEEE Trans. Commun.*, 1339–1349, 1989.

[8] C.E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, 27, 379–423 and 623–656, July and Oct. 1948.

[9] D. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, 40, 1098–1101, 1952.

[10] P.W. Jones and M. Rabbani, "*Digital Image Compression Techniques*," SPIE Optical Engineering Press, Bellingham, WA., 60, 1991.

[11] N. Ahmed, T.R. Natarajan, and K.R. Rao, "On image processing and a discrete cosine transform," *IEEE Trans. Comput.*, IT-23, 90–93, Jan. 1974.

[12] J.J. Hwang and K.R. Rao, "*Techniques and Standards For Image, Video, and Audio Coding*," Prentice Hall, Upper Saddle River, NJ., 22, 1996.

[13] R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine*, IT-1, 4–29, April 1984.

[14] W.H. Equitz, "A new vector quantization clustering algorithm," *IEEE Trans. Acous., Speech, Sig. Proc.*, ASSP-37(10), 1568–1575, 1989.

[15] B.G. Haskell and J.O. Limb, "Predictive video encoding using measured subjective velocity," U.S. Patent No. 3,632,865, Jan. 1972.

[16] A.N. Netravali and J.D. Robbins, "Motion –compensated television coding: Part I," *Bell Syst. Tech. J.*, 58, 631–670, March 1979.

[17] J.R. Jain and A.K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, COM-29, 1799–1808, Dec. 1981.

[18] T. Koga et al., "Motion compensated interframe coding for video conferencing," *NTC '81, National Telecommun. Conf.*, G5.3.1–G5.3.5, New Orleans, LA., Nov. 1981.

[19] ITU-T SG 15 WP 15/1, *Draft Recommendation H.263 (Video coding for low bitrate communications)*, Document LBC-95-251, Oct 1995.

[20] M. Roser et al., "Extrapolation of a MPEG-1 video-coding scheme for low-it-rate applications," *SPIE Video Commun. And PACS for Medical Appl.*, Berlin, Germany, 1977, 180–187, April 1993.

[21] CCITT SG XV WP/1/Q4 Specialist Group on Coding for Visual Telephony, *Description of Ref. Model 6* (RM6), Document 396, Oct. 1988.

[22] CCITT SG XV WP/1/Q4 Specialist Group on Coding for Visual Telephony, *Description of Ref. Model 8* (RM8), Document 525, June 1989.