Scanning the Technology

On the Applications of Multimedia Processing to Communications

RICHARD V. COX, FELLOW, IEEE, BARRY G. HASKELL, FELLOW, IEEE, YANN LECUN, MEMBER, IEEE, BEHZAD SHAHRARAY, MEMBER, IEEE, AND LAWRENCE RABINER, FELLOW, IEEE

Invited Paper

The challenge of multimedia processing is to provide services that seamlessly integrate text, sound, image, and video information and to do it in a way that preserves the ease of use and interactivity of conventional plain old telephone service (POTS) telephony, irrelevant of the bandwidth or means of access of the connection to the service. To achieve this goal, there are a number of technological problems that must be considered, including:

- *compression and coding of multimedia signals, including algorithmic issues, standards issues, and transmission issues;*
- *synthesis and recognition of multimedia signals, including speech, images, handwriting, and text;*
- *organization, storage, and retrieval of multimedia signals, including the appropriate method and speed of delivery (e.g., streaming versus full downloading), resolution (including layering or embedded versions of the signal), and quality of service, i.e., perceived quality of the resulting signal;*
- *access methods to the multimedia signal (i.e., matching the user to the machine), including spoken natural language interfaces, agent interfaces, and media conversion tools;*
- *searching (i.e., based on machine intelligence) by text, speech, and image queries;*
- *browsing (i.e., based on human intelligence) by accessing the text, by voice, or by indexed images.*

In each of these areas, a great deal of progress has been made in the past few years, driven in part by the relentless growth in multimedia personal computers and in part by the promise of broad-band access from the home and from wireless connections. Standards have also played a key role in driving new multimedia services, both on the POTS network and on the Internet.

It is the purpose of this paper to review the status of the technology in each of the areas listed above and to illustrate current capabilities by describing several multimedia applications that have been implemented at AT&T Labs over the past several years.

Manuscript received June 9, 1997; revised December 3, 1997. The Guest Editor coordinating the review of this paper and approving it for publication was T. Chen.

The authors are with the Speech and Image Processing Services Research Laboratory, AT&T Labs, Florham Park, NJ 07932-0971 USA.

Publisher Item Identifier S 0018-9219(98)03279-4.

Keywords*—*AAC, access, agents, audio coding, cable modems, communications networks, content-based video sampling, document compression, fax coding, H.261, HDTV, image coding, image processing, JBIG, JPEG, media conversion, MPEG, multimedia, multimedia browsing, multimedia indexing, multimedia searching, optical character recognition, PAC, packet networks, perceptual coding, POTS telephony, quality of service, speech coding, speech compression, speech processing, speech recognition, speech synthesis, spoken language interface, spoken language understanding, standards, streaming, teleconferencing, video coding, video telephony.

I. INTRODUCTION

In a very real sense, virtually every individual has had experience with multimedia systems of one type or another. Perhaps the most common multimedia experiences are reading the daily newspaper or watching television. These may not seem like the exotic multimedia experiences that are discussed daily in the media or on television, but nonetheless, these are multimedia experiences.

Before proceeding further, it is worthwhile to define exactly what constitutes a multimedia experience or a multimedia signal so we can focus clearly on a set of technological needs for creating a rich multimedia communications experience. The dictionary definition of multimedia is:

> including or involving the use of several media of communication, entertainment, or expression.

A more technological definition of multimedia, as it applies to communications systems, might be the following:

> integration of two or more of the following media for the purpose of transmission, storage, access, and content creation:
>
> - text;
> - images;
> - graphics;
> - speech;

- audio;
- video;
- animation;
- handwriting;
- data files.

With these definitions in mind, it should be clear that a newspaper constitutes a multimedia experience since it integrates text and halftone images and that television constitutes a multimedia experience since it integrates audio and video signals. However, for most of us, when we think about multimedia and the promise for future communications systems, we tend to think about movies like *Who Framed Roger Rabbit?* that combine video, graphics, animation with special effects (e.g., morphing of one image to another) and compact disc (CD)-quality audio. On a more business-oriented scale, we think about creating virtual meeting rooms with three-dimensional (3-D) realism in sight and sound, including sharing of whiteboards, computer applications, and perhaps even computer-generated business meeting notes for documenting the meeting in an efficient communications format. Other glamorous applications of multimedia processing include:

- distance learning, in which we learn and interact with instructors remotely over a broad-band communication network;
- virtual library access, in which we instantly have access to all of the published material in the world, in its original form and format, and can browse, display, print, and even modify the material instantaneously;
- living books, which supplement the written word and the associated pictures with animations and hyperlink access to supplementary material.

It is important to distinguish multimedia material from what is often referred to as multiple-media material. To illustrate the difference, consider using the application of messaging. Today, messaging consists of several types, including electronic mail (e-mail), which is primarily text messaging, voice mail, image mail, video mail, and handwritten mail [often transmitted as a facsimile (fax) document]. Each of these messaging types is generally (but not always) a single medium and, as such, is associated with a unique delivery mechanism and a unique repository or mailbox. For convenience, most consumers would like to have all messages (regardless of type or delivery mechanism) delivered to a common repository or mailbox—hence the concept of multiple media's being integrated into a single location. Eventually, the differences between e-mail, voice mail, image mail, video mail, and handwritten mail will disappear, and they will all be seamlessly integrated into a true multimedia mail system that will treat all messages equally in terms of content, display mechanism, and even media translation (converting the media that are not displayable on the current access device to a medium that is displayable, e.g., text messages to voice messages for playback over a conventional telephone, etc.).

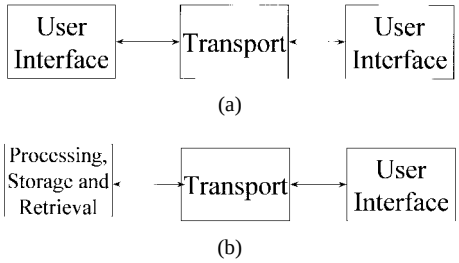

Fig. 1. Elements of multimedia systems used in (a) person-to-person and (b) person-to-machine modes.

A. Elements of Multimedia Systems

There are two key communications modes in which multimedia systems are generally used, namely, person-to-person (or equivalently people-to-people) communications and person-to-machine (or equivalently people-to-machine) communications. Both of these modes have a lot of commonality, as well as some differences. The key elements are shown in Fig. 1.

In the person-to-person mode, shown in Fig. 1(a), there is a user interface that provides the mechanisms for all users to interact with each other and a transport layer that moves the multimedia signal from one user location to some or all other user locations associated with the communications. The user interface has the job of creating the multimedia signal, i.e., integrating seamlessly the various media associated with the communications, and allowing users to interact with the multimedia signal in an easy-to-use manner. The transport layer has the job of preserving the quality of the multimedia signals so that all users receive what they perceive to be high-quality signals at each user location. Examples of applications that rely on the person-to-person mode include teleconferencing, video phones, distance learning, and shared workspace scenarios.

In the person-to-machine mode, shown in Fig. 1(b), there is again a user interface for interacting with the machine, along with a transport layer for moving the multimedia signal from the storage location to the user, as well as a mechanism for storage and retrieval of multimedia signals that are either created by the user or requested by the user. The storage and retrieval mechanisms involve browsing and searching (to find existing multimedia data) and storage and archiving to move user-created multimedia data to the appropriate place for access by others. Examples of applications that rely on the person-to-machine mode include creation and access of business meeting notes, access of broadcast video and audio documents and performances, and access to video and document archives from a digital library or other repositories.

B. Driving Forces in the Multimedia Communications Revolution

Modern voice communications networks evolved around the turn of the twentieth century with a focus on creating *universal service*, namely, the ability to automatically connect any telephone user with any other telephone user without the need for operator assistance or intervention.

This revolutionary goal defined a series of technological problems that had to be solved before the vision became reality, including the invention of the vacuum tube for amplification of telephone signals, mechanical switching to replace the operator consoles that were used in most localities, numbering plans to route calls, signaling systems to route calls, etc. The first transcontinental call in the United States was completed in 1915, thereby ushering in the "modern age of voice communications," an age that has been developed and improved upon for the past 80 or so years.

We are now in the midst of another revolution in communications, one that holds the promise of providing ubiquitous service in multimedia communications. The vision for this revolution is to provide seamless, easy-to-use, high-quality, affordable multimedia communications between people and machines anywhere and anytime. There are three key aspects of the vision that characterize the changes that will occur in communications once this vision is achieved.

1) The basic currency of communications evolves from narrow-band voice telephony to seamlessly integrated, high-quality, broad-band transmission of multimedia signals.
2) The basic access method changes from wireline connections to combinations of wired and wireless, including cable, fiber, cell sites, satellite, and even electrical power lines.
3) The basic mode of communications expands from primarily involving people-to-people communications to include people-to-machine communications.

There are a number of forces that are driving this multimedia revolution, including:

- the evolution of communications networks and data networks into today's modern plain old telephone service (POTS) network and packet (including the Internet) networks, with major forces driving these two networks into an integrated structure;
- the increasing availability of (almost unlimited) bandwidth on demand in the office, the home, and eventually on the road, based on the proliferation of high-speed data modems, cable modems, hybrid fiber-coax systems, and, recently, a number of fixed wireless access systems;
- the availability of ubiquitous access to the network via local-area networks (LAN's), wireline, and wireless networks providing the promise of anywhere, anytime access;
- the ever increasing amount of memory and computation that can be brought to bear on virtually any communications or computing system—based on Moore's law of doubling the communications and memory capacity of chips every 18 or so months;
- the proliferation of smart terminals, including sophisticated screen phones, digital telephones, multimedia personal computers (PC's) that handle a wide range of text, image, audio, and video signals, "network computers" and other low-cost Internet access terminals, and personal digital assistants (PDA's) of all types that are able to access and interact with the network via wired and wireless connections;
- the digitization of virtually all devices, including cameras, video capture devices, video playback devices, handwriting terminals, sound capture devices, etc., fueled by the rapid and widespread growth of digital signal-processing architectures and algorithms, along with associated standards for plug-and-play as well as interconnection and communications between these digital devices.

C. Technology Aspects of Multimedia Systems

For multimedia systems to achieve the vision of the current communications revolution and become available to everyone, much as POTS service is now available to all telephony customers, a number of technological issues must be addressed and put into a framework that leads to seamless integration, ease of use, and high-quality outputs. Among the issues that must be addressed are the following:

- the basic techniques for compressing and coding the various media that constitute the multimedia signal, including the signal-processing algorithms, the associated standards, and the issues involved with transmission of these media in real communications systems;
- the basic techniques for organizing, storing, and retrieving multimedia signals, including both downloading and streaming techniques, layering of signals to match characteristics of the network and the display terminal, and issues involved with defining a basic quality of service (QoS) for the multimedia signal and its constituent components;
- the basic techniques for accessing the multimedia signals by providing tools that match the user to the machine, such as by using "natural" spoken language queries, through the use of media conversion tools to convert between media, and through the use of agents that monitor the multimedia sessions and provide assistance in all phases of access and utilization;
- the basic techniques for searching in order to find multimedia sources that provide the desired information or material—these searching methods, which in essence are based on machine intelligence, provide the interface between the network and the human user and provide methods for searching via text requests, image matching methods, and speech queries;
- the basic techniques for browsing individual multimedia documents and libraries in order to take advantage of human intelligence to find desired material via text browsing, indexed image browsing, and voice browsing.

D. Illustrative Examples of Multimedia Processing Systems

To bring the various technology aspects of multimedia systems into focus, we include a number of illustrative

examples of multimedia processing systems in this paper. Our goal here is to show how technology and practice come together in different systems to create interesting combinations of networking, computing, and communications. We believe these systems provide a good picture of where we anticipate multimedia communications systems to be heading in the next decade or so.

The individual multimedia processing systems we will discuss include the following:

- teleconferencing systems, which integrate voice, video, application sharing, and data sharing;
- Internet services, which integrate the POTS and packet networks to provide a wide range of new services with integrated control and signal processing;
- FusionNet service, which exploits Internet control of content and video cassette recorder (VCR)-like features for viewing the video and provides guaranteed QoS POTS access to video content through standard Internet service providers (ISP's), who provide integrated services digital network (ISDN) connectivity;
- the CYBRARY digital-library project, which aims to provide a digital-library experience that is "better than being there live";
- the Pictorial Transcripts system, which automatically organizes, condenses, and indexes video programs into a digital multimedia library and makes them available for searching, browsing, and selective retrieval over low-bandwidth communications networks.

In the next several sections, we expand further on the driving forces that are shaping the multimedia communications revolution and the technologies that will determine the shape and the user interfaces to multimedia systems. Then we discuss the representative applications listed above and summarize our views as to how this multimedia communications revolution will unfold over time.

II. The Multimedia Communications Revolution

The multimedia experience represents the convergence of computing, communications, and information sciences. For the consumer to reap the maximum benefit from the multimedia revolution, there are a number of factors that must come together in order to provide a seamless infrastructure and seamless access, display, storage, and processing capabilities. In the following sections, we review the status of several key technologies and products and show how they influence progress toward making multimedia systems more broadly available.

A. The Evolution of Communications Networks

For more than 100 years, the POTS network [as shown in Fig. 2(a)] has been the primary focus of conventional voice-band communications [1]. The POTS network is well designed and well engineered for the transmission and switching of 3-kHz voice calls. The network is a real-time, low-latency, high-reliability, moderate-fidelity, voice telephony network. Since its initial design, there have been

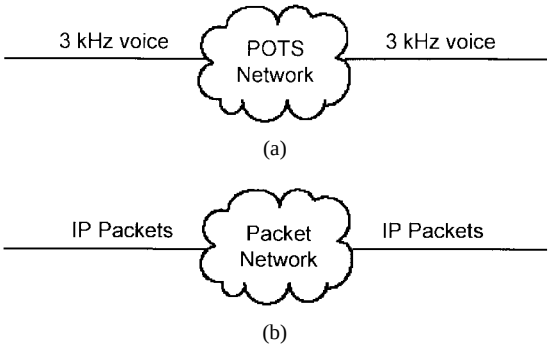

Fig. 2. The basic (a) POTS and (b) packet networks.

a number of key architectural modifications, the most significant of them being the addition of an independent digital signaling system [signaling system (SS) 7] for faster call setup and the digitization of the network into 64 kb/s pipes (for higher reliability and improved fidelity). The POTS network is not designed, nor is it especially well suited, for other forms of communications, including wide-band speech, audio, images, video, facsimile, and data. Hence, these other communication signals have been delivered over the POTS network through the use of telephony modem connections.[1] Higher data rate connections to the POTS network can be obtained by using ISDN to access two or more 64-kb/s channels (the so-called B channels) or via digital subscriber line (DSL) technology, such as asymmetric (A)DSL or hybrid (H)DSL, which bypass the local switch digitization hardware and provide direct, high-data-rate access to the network.

The POTS network is inherently "telephone" or "handset" oriented and is driven by the needs of real-time voice telephony. There are approximately 270 million users of the POTS network in the United States, making POTS access essentially ubiquitous. On the other hand, the POTS network has high access costs (paid to the local access providers and mandated by legislation in the United States) and, for international calls, high settlement costs (paid to international postal, telephone, and telegraph organizations). The modern POTS network looks more like that shown in Fig. 3, with several devices connected to the network via modems, than the simple view shown in Fig. 2(a).

About 30 years ago, a second communications network was created with the goal of providing a better transport mechanism for data networking. The resulting network, shown in Fig. 2(b), has evolved and is called a packet network because data is transmitted and routed along the network in the form of Internet protocol (IP) packets. Packet networks are general-purpose data networks that are not tied to fixed-bandwidth circuits. Instead, they are designed to transmit bits (in the form of packets of fixed or variable length) only when there are bits to transmit. Packet networks evolved independently of telephone networks for the purpose of moving bursty,

[1] This often leads to the curious necessity to convert an inherently digital signal, e.g., data signals, to analog form, via the voice-band modem, so that it can be redigitized and reformatted by the telephony modem in the local switching office.

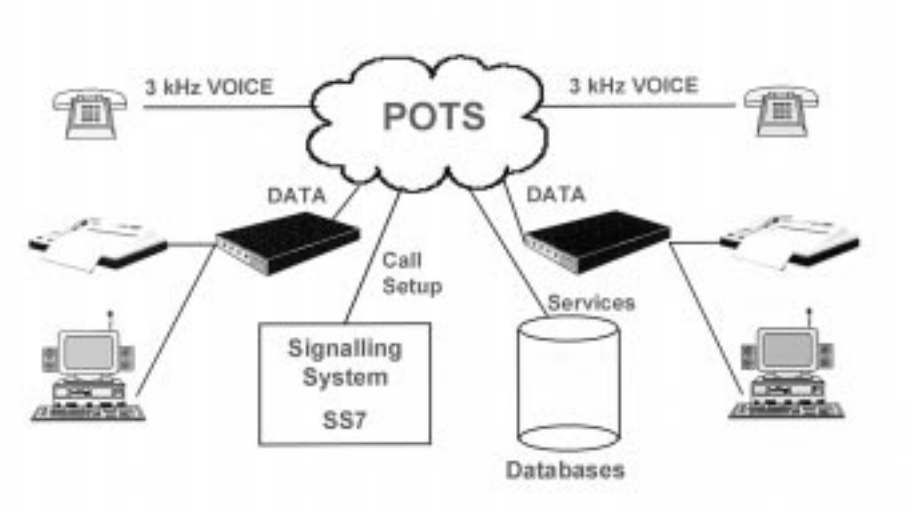

Fig. 3. The POTS network with signaling, service-oriented data bases, and connections for facsimile devices and PC's via voice-band modems.

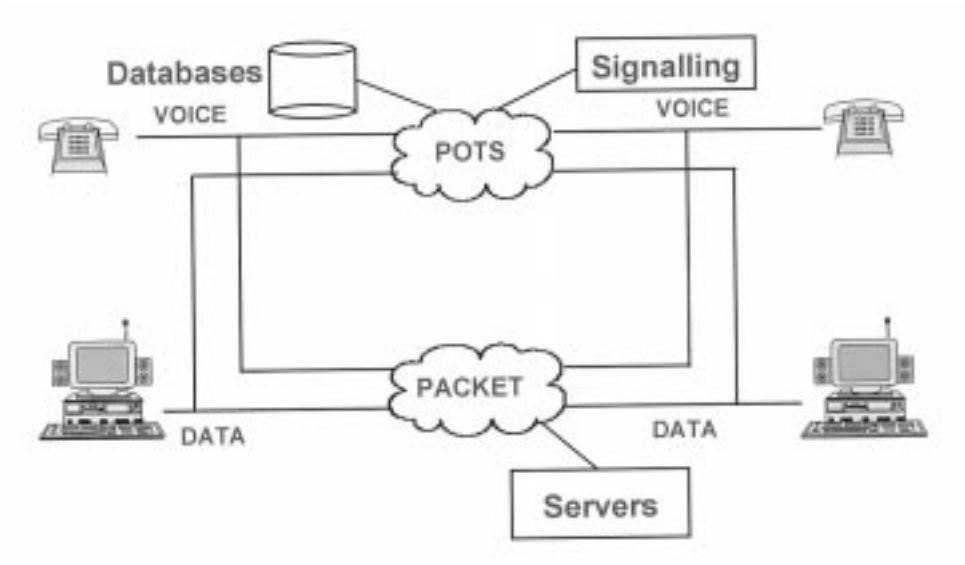

Fig. 4. The telecommunications network of today.

nonreal-time data among computers and are distinguished by the property that packet communications are routed by address information contained in the data stream itself. The packet network is especially well suited for sending data of various types, including messages, facsimile, and still images. The network is not well suited for sending real-time communication signals, however, such as speech, audio, and video. The packet network is primarily accessed by client programs in PC's, and so it is inherently PC oriented and client/server driven. It provides access to distributed data bases and, via some excellent search engines, has excellent search capabilities.

There are approximately 30 million users of the packet network in the United States, a number that is growing rapidly and will continue to do so over the next decade. Today, the Internet (the largest of the existing packet networks) connects more than 40 million computers in some 140 countries. The growth rate is astounding, with anticipated revenue from content-related Internet services expected to exceed $10 billion by the end of the decade.

A key problem with packet networks is that they were not designed to handle voice traffic. With the emergence of the multimedia PC and its concomitant signal-processing capability, a simple solution was proposed, namely, the compression (and decompression) and coding (and decoding) of voice into (from) IP packets using the PC processor and the sound card inherent in the multimedia PC. Unfortunately, associated with this solution was the problem of long delays due to several factors, including:

- the algorithmic processing delay required for low-bit-rate compression (on the order of 25–65 ms for high-quality compression algorithms);
- the algorithmic processing and delay for acoustic echo cancellation associated with sound transmission from the PC loudspeaker to the PC microphone (on the order of 60 ms for good echo cancellation);

- packet assembly and protocol processing [on the order of 24 ms for speech sampled at 16 kb/s or 48 ms for speech sampled at 8 kb/s over asynchronous transmission mode (ATM) networks];
- the packet buffering that is required to match the packet delivery mechanism with the packet network (on the order of 20–80 ms);
- modem delays (on the order of 35–200 ms/connection pair for V.34 modems);
- routing/queuing delays inherent at each router within the packet network (on the order of hundreds of milliseconds and more);
- propagation delays (on the order of 5 μs/mile on fiber or up to 120 ms for a round trip of 25 000 miles around the world);
- network congestion (including both lost and delayed packets) due to traffic in the packet network;
- the delay associated with the PC sound card

There are several clever signal-processing techniques that have been applied to overlap delays from the coder, echo canceler, packet assembly, and packet buffering so that the overall delay is not the sum of the individual components of the delay. However, the major bottleneck in transmission over packet networks is the networking component, which cannot be reduced significantly until some QoS protocol, such as the resource reservation protocol (RSVP) or IPv6, is applied, which provides for virtual circuits to minimize the delay in determining the route for packets (i.e., predetermined circuits) and priority queues so that real-time voice traffic takes priority over data traffic at each switch and router in the network. Use of the ATM protocol will also reduce delay because of the small packet size (48 bytes) and the small header size (5 bytes).

The reason that delay is so important in packet telephony is that when the round-trip delay in any network becomes too large, it affects the perceived quality of the connection. Moreover, if there is a perceptible echo as well, the perceived quality is even poorer. The effects of long delay and echoes are measurable. Based on extensive human perception studies, the International Telecommunications Union—Telecommunications Sector (ITU-T) has concluded in Recommendation G.114 that round-trip delays of less than 300 ms, with no echo, are imperceptible to even the most sensitive users and applications. At round-trip delays of 500 ms, a 20–30% loss in conversational efficiency (appropriately measured) is observed [2]. In another study, with round-trip delays of 600 and 1200 ms, 35–45% of the phone connections were rejected as unusable by actual telephone customers. At delays as large as 1200 ms, it becomes difficult to interrupt the speaker, and therefore the conversational dynamics become that of a "push-to-talk" or half-duplex-type system rather than the usual "interrupt-driven" system.

If there are echoes present, they must be canceled in order to preserve conversational dynamics. The effect of echoes is measurable for round-trip delays as low as 1.5 ms [4]. The amount of echo cancellation needed depends on the round-trip delay of the echo and the noise mixed in with the original signal. (In conventional telephony, this noise is usually assumed to be line noise, but for the purposes of multimedia communications, it could be due to high levels of background noise or lossy channels as well. No definitive study has been made that includes all of these potential noise sources.) The greater the delay, the greater the amount of echo cancellation that is needed. A general rule, used in conventional telephony, is that echoes must be canceled to the point where they are inaudible compared with the line noise associated with the connection. When the echoes are not adequately canceled, the louder they are, the more disconcerting they become for the speaker. The combination of a loud echo with a fairly long delay can render the system practically unusable because of its psychological impact on most speakers.

1) Today's Communications Network: The evolution of the POTS and packet networks has led to the modern communications network shown in Fig. 4. As discussed above, the POTS network is a connection-oriented, narrow-band, voice-centric network whose main functionality is digital switching and transmission of 3-kHz voice signals (from a standard telephone handset) digitized to 64-kb/s digital channels. Data (in the form of modem signals from a PC or a fax machine) are handled by the POTS network through the use of a voice-band modem, which limits the speed of transmission to rates below 64 kb/s. Services on the POTS network (e.g., basic long distance, 800-number calling, call forwarding, directory services, conferencing of calls, etc.) are handled via a distributed architecture of circuit switches, data bases, and switch adjuncts. Signaling (for call setup and call breakdown, look ahead, routing, data base lookup, and information passing) is handled by a side (out of band) digital channel (the so-called SS7), which is essentially a parallel 64-kb/s digital network.

As discussed above, the packet network is a connectionless (in the sense that there is no transmission setup), wide-band, data-centric network whose main functionality is routing and switching data from one location to another in the network using the standard transmission protocol, transmission control protocol (TCP)/IP, associated with the Internet. The packets consist of a header and a payload, where the header contains information about the source and destination addresses of the packet and the payload is the actual data being transmitted. Services in the packet network are provided by servers attached to the packet network and running in a client-server mode. Typical services include browsing, searching, access to newspapers and magazines, access to directories, access to bookstores, stock offerings, etc. Since packets are self-routing, there is no outside signaling system associated with packet networks.

A major problem in today's network is that because of their separate evolution, the POTS and packet networks are only weakly coupled at best. Hence, services that are available on the POTS network cannot be accessed from a PC connected to the packet network; similarly, services that are available on the packet network cannot be

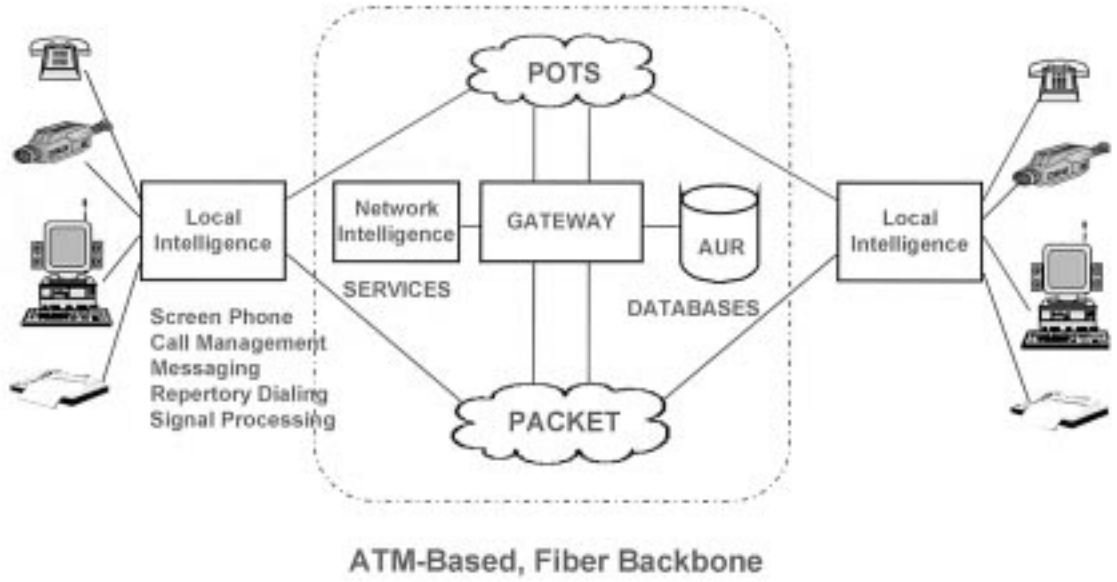

Fig. 5. The telecommunications network of tomorrow.

accessed from a telephone connected to the POTS network. This is both wasteful and inefficient, since essentially identical services are often offered over both networks (e.g., directory services, call centers, etc.), and such services need to be duplicated in their entirety.

Recently, however, telecommunications networks have begun to evolve to the network architecture shown in Fig. 5. In this architecture, which already exists in some early instantiations, there is tight coupling between the POTS and packet networks via a gateway, which serves to move POTS traffic to the packet network and vice versa. Intelligence in the network is both local (i.e., at the desktop in the form of a PC, a screen phone, a personal information manager, etc.) and distributed throughout the network in the form of active data bases (e.g., an active user registry of names and reach numbers). Services are implemented by being attached to the gateway between the POTS and packet networks. In this manner, any given service is, in theory, accessible over both the POTS and packet networks and from any device connected to the network.

The key features of the integrated communications network of Fig. 5 are the following.

- The existing POTS and packet networks remain essentially intact while leveraging the outstanding strengths of each network.
- Each network can control the other, in a synergistic way, so as to provide the mechanism for a broad new set of multimedia services.
- New multimedia services can be created that provide significant improvements in ease of use, convenience, ubiquity, etc. with no perceived loss in QoS for any signal in the network.

Some examples of multimedia services that will evolve over the enhanced communications network include:

- packet network point-and-click interfaces for initiation of POTS network multiparty conference calls, as well as POTS multimedia messaging services;
- packet network call scheduling for POTS network calls to service bureaus and help lines;
- packet network requests for remote medical services, with voice and video over POTS and image and data over the packet network;
- packet network ordering of seats for sporting events, shows, opera, etc., with POTS network views (simulated) from the selected seats.

B. Bandwidth on Demand

Conventional access to the POTS network is limited to less than a 64-kb/s data rate because of the restriction to using voice-band modems. As will be shown later in this paper, multimedia signals of reasonable quality generally require significantly higher data rates. This has led to a dilemma for telecom providers, namely, how to interface customers with the ever growing set of broad-band services that will be provided over time.

To set the stage for this discussion, Fig. 6 shows a plot of data rates that are available for various types of commercial transmission and access over time. There are three distinct sets of data shown in this plot, namely, the maximum capacity for moving broad-band data over telecommunications networks (usually via fiber-optic techniques along the backbone networks of telecommunications), the maximum capacity for moving telecom traffic over so-called wide-area networks (WAN's) via standard switched telecom lines, and the maximum capacity for moving data over so-called LAN's, which typically interconnect computers, routers, and switches in a local data communications environment. It can be seen that the data rates associated

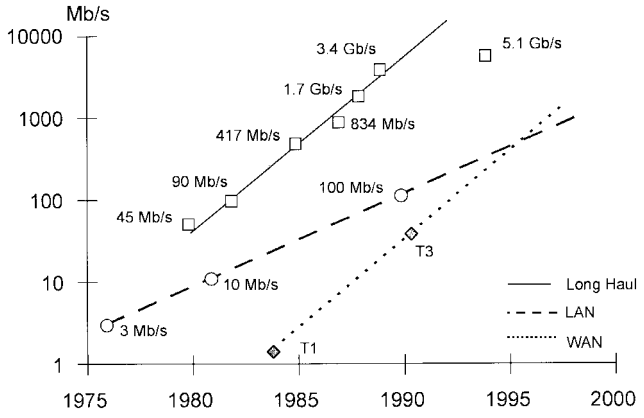

Fig. 6. Plots of transport capability of voice and data systems over the past 20 years.

with all three of these transport modes have risen steadily over time, typically from the megabit/second range to the multimegabit/second or even the gigabit/second range over a 10–20-year period.

The access picture changes dramatically when we consider the consumer who tries to access the POTS and packet networks from either the home or outdoors environments. Fig. 7 shows a similar plot of transport capability of the LAN, the WAN, and consumer access over time. The first two sets of points are the same ones shown in Fig. 6. The dramatic point shown in this figure is the failure of consumer access to keep up with commercial access. This is due primarily to the necessity to access the "network" via voice-band modems. These modems have shown dynamic growth from 1200-b/s capability in the 1970's to 56-kb/s capability in 1997, a 50-to-1 growth rate over the past two decades. What is painfully clear, however, is the difference between the commercial access rates of 10–100 Mb/s that are available in virtually all businesses today and the 50-kb/s rates that are available to consumers in the home and via wireless access. Such low access data rates will not support the types of services that consumers will demand as the multimedia revolution sweeps across the telecom and data landscapes. As shown in Fig. 7, we believe that there are a number of technologies that will effect a discontinuity in the curve of consumer access data rates and provide access rates of from 1.5 Mb/s (T1 rates) up to 155 Mb/s (OC3 rates) to consumers in both their homes and in outdoor wireless environments. These include the following.

- *XDSL modems:* the copper loop between the subscriber premises and the telephone central office (CO) can be utilized to provide high-speed data access to and from the home through the use of special DSL modems at each end of the connection and through the use of line conditioning by the local exchange carrier (LEC). There are three popular forms of XDSL, namely:

  1) ADSL, which operates at distances up to 18 K-ft from the subscriber to the CO and provides downstream data rates of 0.6–0.8 Mb/s and upstream data rates of 64–640-kb/s over a single twisted pair of copper wires;
  2) HDSL, which operates at distances up to 10 K-ft from the subscriber to the CO and provides data rates of up to 1.5 Mb/s in both directions over two twisted pairs of copper wires;
  3) Symmetric DSL, which operates at distances of up to 8 K-ft from the subscriber to the CO and provides data rates of up to 1.5 Mb/s in both directions over a single twisted pair of copper wires.

  A new DSL technology, called self-adaptive (SA)DSL, attempts to solve the problem of needing line conditioning on a line-by-line basis by using signal-processing methods to condition the subscriber line (or lines) adaptively and automatically.
- *Cable modems:* recently, several companies have begun experimenting with one-way and two-way delivery over the cable that comes into a consumer's home as part of cable television (CATV) service. One-way delivery of service utilizes a single 6-MHz cable channel for downstream delivery of service at rates of 1–10 Mb/s from a cable head end delivery system (usually an ISP with a connection to the cable head end) and uses a conventional voice modem for upstream delivery at conventional POTS modem rates (typically 28.8–56 kb/s). The AT&T SAIL System is one such system that is undergoing tests within 100 homes in the New Jersey area. Two-way cable delivery of services utilizes two 6-MHz cable channels: one for downstream delivery of service and one for upstream delivery of service. Such two-way cable systems need special amplifiers and isolation systems to operate properly, since the cable used for CATV systems was designed for one-way delivery of service and is notoriously bad for upstream delivery without special hardware. The CATV companies are all committed to upgrading their cable plants to provide two-way delivery over time.
- *Fiber-to-the-home (FTTH):* one very simple way of significantly increasing the bandwidth (to rates on the order of OC3 or 155 Mb/s) to the home is to replace the copper cable with optical fiber. This option is very costly because of the need to convert from electrical signals (in the home) to optical signals (required for transmission over fiber). Early experience with fiber systems indicates that the cost of both bringing the fiber into the home and conversion between electrical and optical signals is still too high to justify broad utilization. As multimedia services start to proliferate, however, and as more consumers take advantage of the benefits of broad-band access to telecom networks, the costs and the needs will make FTTH economically viable over time. These systems are currently undergoing limited trial and evaluation in customer homes.
- *Hybrid fiber-coax:* an alternative to bringing fiber to the home is to bring the fiber to a central location

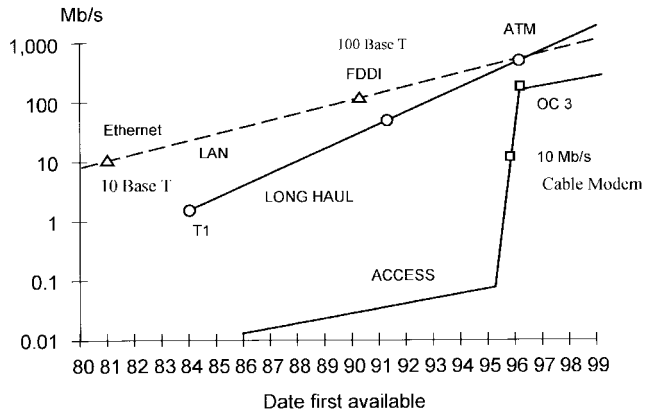

Fig. 7. Consumer and commercial access data rates versus time of initial introduction.

in each neighborhood (e.g., the curb) and then do the conversion from optical to electrical with delivery of high-data-rate electrical signals via standard coaxial cable. Such a system has economic benefits (a single fiber connection can often serve up to 2000 homes) and provides a reasonable tradeoff between cost and data rates provided to the home.

- *Fixed wireless loops:* yet another interesting alternative is to provide in the home a wireless local station that is fixed in location and therefore can be positioned and pointed directly at the network wireless base station. Such an arrangement has the advantage that it can maintain a much higher signal-to-interference ratio than a mobile wireless station and therefore will be robust to fading, traffic, multipath interference, etc. Furthermore, by utilizing the available radio bandwidth more efficiently than a mobile system (since user locations are known precisely), these fixed wireless systems can provide moderate-to-high data rates reliably and cost effectively. Early proposals for these systems project data rates from 128 kb/s to upwards of 155 Mb/s for high-powered systems.

C. Ubiquitous Access

A key aspect of the multimedia revolution is ubiquitous access to information and services anywhere and anytime. The concept of ubiquitous access to people and to standard telephony services (e.g., directory assistance, call conferencing, etc.) via the conventional telephone was a cornerstone of the universal access revolution associated with the POTS network. However, as the packet network has evolved, the meaning of ubiquitous access has become less clear because of the availability of broad-band services over the Internet and the ability to interact with Web-based services via the standard graphical user interface (GUI) associated with PC's.

With the advent of wireless communications, PDA's, pagers, etc. for both voice and data, new meaning has been attached to the concept of ubiquitous access. Certainly, the POTS network has become available around the clock and virtually anywhere. However, ubiquitous access to broad-band services would seem to require a broad-band access terminal (e.g., a portable PC with wireless broad-band access), and that indeed is one potential solution that is actively being pursued both in research and among cellular service providers. There is another way of providing ubiquitous access to the service concepts associated with Web-based services, however, and that is via voice access to what are conventionally considered to be Web services, e.g., voice-based intelligent agents that access Web information servers to retrieve information about traffic, the weather, sports scores, stock price quotations, theater bookings, etc. With the linking of the POTS and packet networks, as envisioned in the future telecom network of Fig. 5, the ability to access packet services from POTS connections becomes simple and straightforward and will enable at least a limited form of ubiquitous access to all services, including the new multimedia services associated with the packet network.

D. Decreased Cost of Computation and Memory

One of the strongest drivers of the multimedia communications revolution is Moore's law, which states that the cost of computation and storage decreases by a factor of two every 18 months. Integrated circuit development has followed Moore's law for both processors and memory for more than 20 years, and there seems to be no technological end in sight for the next 10–20 years. However, the financial costs of setting up the fabrication lines to actually build the resulting very-large-scale-integration (VLSI) chips remains an important consideration in the growth of integrated circuits over this time frame.

The implications of Moore's law for multimedia processing become clear when one examines the growth, over time, of processing power [measured in millions of instructions per second (MIPS)] (Fig. 8) and storage (measured in bits/chip) (Fig. 9). Two curves are shown in Fig. 8, namely, the processing power of specialized digital signal processing (DSP) chips (the upper curve) and the processing power of standard processors (e.g., the X86 series from Intel). It can be seen that the processing power of DSP chips was sufficient for some speech-processing applications (10–50 MIPS) as early as 1984 and for some image and video processing applications (100–1000 MIPS) as early as 1990. More important, perhaps, is the fact that the processing capability of standard PC processors was sufficient for some speech applications by 1988 and for some image and video applications by 1994. Hence, multimedia applications started to appear in software on the desktop by the mid-1990's and, with the appearance and widespread acceptance of sound cards and accelerated video cards, proliferated rapidly over the past few years. Most recently, with the growth in the use of the Intel MMX series processor, advanced multimedia applications have started to become popular on the desktop.

Fig. 9 shows a similar trend in the information storage capacity of solid-state memory chips. It can be seen that as early as 1980, with the widespread acceptance of the 64-

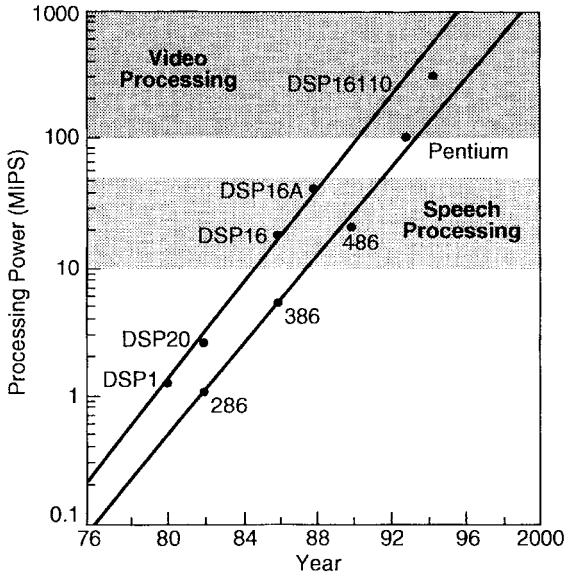

Fig. 8. Processing power on a single VLSI chip, over time, for both DSP and processor chips.

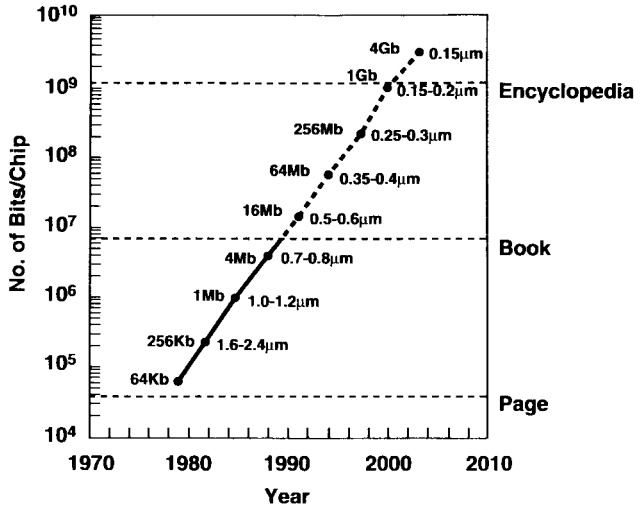

Fig. 9. Memory storage capacity of single VLSI chips over time.

Kb memory chip,[2] there was sufficient memory capacity of a single chip to support text-processing applications of pages of text at a time. By 1990, with the appearance of the 16-Mb memory chip, the contents of an entire book could be stored in a single memory chip, thus expediting advanced searching, sorting, indexing, and text-browsing applications, as well as presaging the beginning of the era of speech- and image-processing applications. By the year 2000, it is forecast that there will be single memory chips that support 1 Gb of memory—enough to store the complete contents of a standard encyclopedia or more than enough to buffer audio and video frames for a complete multimedia processing system.

[2]The computer industry often uses two definitions of the term "kilo," namely k = 1000 and K = 1024. They also use the notation b = bit and B = byte.

E. Smart Terminals

It has been said that every major advance in networking has been preceded by an advance in the user interface that has precipitated the acceptance and growth of the networking advance. By way of example, the invention of the telephone preceded the growth of switch networks, the invention of the television preceded the growth of TV networks and CATV, the radio telephone led to the cellular network, the PC led to the LAN/WAN network, and the browser led to the phenomenal growth of the Internet and the World Wide Web. With this background, it should be clear that for the multimedia revolution to flourish, there need to be new smart terminals created in order to facilitate the creation, display, access, indexing, browsing, and searching of multimedia content in a convenient and easy-to-use manner. Such terminals may be the outgrowth of today's cellular phones combined with massively available PDA's, or they may evolve from some entirely new efforts to build small, portable, wireless display terminals with speech control. Such a terminal would need a much larger video display than those associated with current cell phones and PDA's in order to display an image or video sequence with any decent resolution.

Fig. 10 shows the history of telephony-based terminal devices that have been created over the past 70 years. The rotary-dial telephone was introduced in 1919 and was the standard terminal device for the telephone network (POTS) until 1963, when the first touch-tone telephones were introduced. The data phone was introduced in 1957, thereby enabling the creation of a separate data network for handling the nonreal-time traffic that began to flow over the switched telephone network. This separate data network evolved into packet networks, of which the Internet is the most popular and most widely used today. The cellular phone was introduced in 1978, the cordless phone in 1983, and the ISDN phone in 1985. Each of these phones precipitated a change in the way consumers utilized services on the POTS network. The first multimedia telephony terminal (that was widely available commercially) was the ISDN video phone, introduced in 1990, followed by the POTS videophone in 1992.[3] Several companies introduced nonstandard video phones in the 1980's, including PictureTel and CLI.

F. ATM and the Age of Digital Communications

One of the biggest factors in the emergence of multimedia computing and processing was the digitization of the telecommunications network, which was completed by the late 1980's. This presaged a new era in communications, where digital representations of both signals and data could interact on a common network, and led to the concepts behind modern data networks and data protocols like ATM [5].

[3]Technically, the first multimedia telephony terminal was the Picturephone that was introduced by the Bell System at the 1964 World's Fair in New York City. The Picturephone was expensive to produce, the proposed service was too costly to support, and therefore the service offering was withdrawn in the early 1970's.

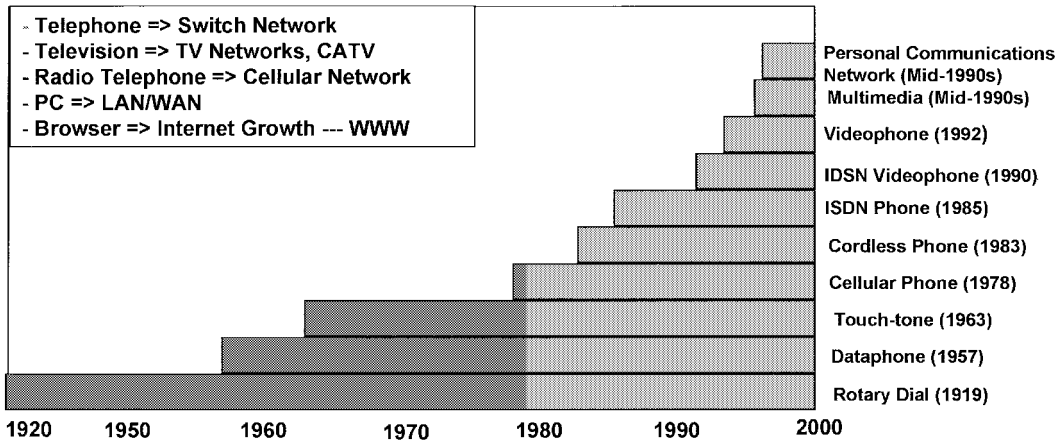

Fig. 10. Growth in the functionality of smart telephone terminals over time.

The ATM protocol is based on the concept of designing a packet protocol that would be appropriate for both real-time signals (e.g., speech, audio, and video) and data. Hence, its packet size is small (53 bytes) for low-latency signal processing and the header size is small (5 bytes) for high efficiency. ATM networks are designed to efficiently switch high-volume voice, audio, and video traffic, yet still maintain their effectiveness for bursty data. The more standard TCP/IP protocol used on the Internet uses significantly larger packets (upwards of 1–2-KB packets) for greater efficiency in moving large, bursty data traffic through the IP network.

III. Key Technologies in Multimedia Processing

In this section, we provide overviews of some of the key technologies that will shape the multimedia revolution. It is important to understand our biases and the point of view we take in describing these technologies. Our first assumption is that multimedia processing is a lot more than signal compression and coding. Therefore, the ability to capture, display, store, compress, and transmit speech, audio, images, video, and handwriting, although necessary, is far from sufficient to generate interesting, addictive, and sticky (i.e., ones that people will use over and over again) multimedia applications.

A second assumption is that multimedia applications can and will thrive only when they are standards based. Thus, for a new coding algorithm to take hold, it must become part of one or more international [e.g., ITU, International Standards Organization (ISO), and Internet Engineering Task Force] standards. No matter how novel or innovative a new signal-processing or multiplexing algorithm might be, without approval and certification by a standards body, proprietary (closed) algorithms will not gain the widespread acceptance they need to fuel new applications.

A third assumption that we make is that the handling of multimedia signals is a key part of the process in any viable multimedia application. By handling of the signal, we mean how it is delivered to the requesting party (e.g., via complete downloading or via some type of streaming procedure) and how it is stored, retrieved, and displayed (e.g., in a single layer or in multiple encodings, with a fixed QoS or with variable QoS depending on network congestion, availability of transmission facilities, etc.).

A fourth, and key, assumption is that the user interface is probably the most important component to the success or failure of a multimedia application. Much as the design of the rotary or touch-tone telephone was essential to the public's acceptance of the dial plan for accessing customers and advanced intelligent services (e.g., interactive voice response using touch-tone entry of responses), the user interface will determine which multimedia applications actually become viable services that customers are willing to pay for and which become bookmarks on an Internet browser that are visited once and never used again. Included in the user interface is the need for voice (or perhaps text) dialogues between the user and the machine (including media conversion when appropriate), the use of synthetic images to support human–machine interactions, and the use of image- and video-recognition technology to access and index documents, convert material to machine readable form via optical character recognition (OCR), find objects in multimedia space, and convert handwritten material to a form that can be readily handled by machine.

The final set of assumptions is that the multimedia experience is often a shared one between two or more people and machines (which sometimes act as intermediaries and sometimes act as manager and guardian of information). As such, there has to be some cooperative agreements between people and machines so as to facilitate both searching (which is essentially based on a combination of human and machine intelligence and knowledge) and browsing (which is essentially based strictly on human intelligence). Our understanding of multimedia searching and browsing is both rudimentary and incomplete. As such, there are a number of key technology challenges that must be tackled and solved before many multimedia applications can be considered truly viable ones. In the remainder of this

Table 1 Characteristics and Uncompressed Bit Rates of Speech, Audio, Image, and Video Signals

Speech/Audio Type	Frequency Range	Sampling Rate	Bits/Sample	Uncompressed Bit rate
Narrowband Speech	200-3200 Hz	8 kHz	16	128 kb/s
Wideband Speech	50-7000 Hz	16 kHz	16	256 kb/s
CD Audio	20-20000 Hz	44.1 kHz	16 x 2 channels	1.41 Mb/s

(a)

Image Type	Pixels per Frame	Bits/Pixel	Uncompressed Size
FAX	1700 x 2200	1	3.74 Mb
VGA	640 x 480	8	2.46 Mb
XVGA	1024 x 768	24	18.87 Mb

(b)

Video Type	Pixels per Frame	Image Aspect Ratio	Frames per Second	Bits/Pixel	Uncompressed Bit rate
NTSC	480 x 483	4:3	29.97	16*	111.2 Mb/s
PAL	576 x 576	4:3	25	16	132.7 Mb/s
CIF	352 x 288	4:3	14.98	12#	18.2 Mb/s
QCIF	176 x 144	4:3	9.99	12	3.0 Mb/s
HDTV	1280 x 720	16:9	59.94	12	622.9 Mb/s
HDTV	1920 x 1080	16:9	29.97	12	745.7 Mb/s

❖ Based on the so-called 4:2:2 color sub-sampling format with two chrominance samples C_b and C_r for every four luminance samples.
\# Based on the so-called 4:1:1 color sub- sampling format with one chrominance samples C_b and C_r for every four luminance samples.

(c)

section, we discuss the state of the technology in each of the key areas outlined above.

A. Compression and Coding of Speech, Audio, Image, and Video Signals

To appreciate the need for compression and coding of the individual signals that constitute the multimedia experience, Table 1 shows the signal characteristics and the resulting uncompressed bit rate necessary to support their storage and transmission with high quality. The table has separate sections for speech and audio signals, images, and video signals, since their characteristics are very different in terms of frequency range of interest, sampling grids, etc.

It can be seen from Table 1(a) that for narrow-band speech, a bit rate of 128 kb/s is required without any form of coding or compression—i.e., twice the rate used in ordinary POTS telephony. For wide-band speech, a bit rate of 256 kb/s is required for the uncompressed signal, and for two-channel stereo-quality CD audio, a bit rate of 1.41 Mb/s is required for the uncompressed signal. We will see later in this section that narrow-band speech can be compressed to about 4 kb/s (a 30-to-1 compression rate), wide-band speech can be compressed to about 16 kb/s (a 15-to-1 compression rate) and CD audio can be compressed to 64 kb/s (a 22-to-1 compression rate) while still preserving the quality of the original signal. Clearly, such high compression ratios are essential for combining speech with audio, images, video, text, and data and transmitting it and storing it efficiently, as required for multimedia processing systems.

Table 1(b) shows the uncompressed size needed for bilevel (fax) and color still images. It can be seen that an ordinary fax of an 8 1/2 by 11 inch document, scanned at 200 dots per inch (dpi), has an uncompressed size of 3.74 Mb, whereas color images (displayed on a computer screen) at VGA resolution require 2.46 Mb and high-resolution XVGA color images require 18.87 Mb for the uncompressed image. It will be shown that most images can be compressed by factors on the order of 100-to-1 (especially text-based fax documents) without any significant loss in quality.

Last, Table 1(c) shows the necessary bit rates for several video types. For standard television, including the North American National Television Systems Committee (NTSC) standard and the European phase alternation line (PAL) standard, the uncompressed bit rates are 111.2 Mb/s (NTSC) and 132.7 Mb/s (PAL). For video-conferencing and video-phone applications, smaller format pictures with lower frame rates are standard, leading to the common intermediate format (CIF) and quarter (Q)CIF standards, which have uncompressed bit rates of 18.2 and 3.0 Mb/s, respectively. Last, the digital standard for high-definition television (HDTV) (in two standard formats) has requirements for an uncompressed bit rate of between 622.9 and 745.7 Mb/s. We will see that modern signal-compression technology leads to compression rates of over 100-to-1.

In the following sections, we review the state of the art in coding and compression of speech/audio, image, and video signals and discuss some of the issues involved in multime-

Table 2 An Illustrative List of Multimedia Applications Using Speech

Multimedia Application	Live conversation?	Real-time network
Video telephony/conference	Yes	Yes
Business conference with data sharing	Yes	Yes
Distance learning/teaching	No	Yes
Single user games	No	Possibly
Multi-user games from remote locations	Possibly	Yes
Multimedia messaging	No	Possibly
Voice annotated documents	No	No

dia implementations of the technology, such as interactions with POTS and packet networks and integration of the technologies for multimedia conferencing applications.

B. Speech Coding for Multimedia Applications

Speech is probably the most natural form of human communication. We begin using it at an early age and, barring some incapacity, we use it for all of our lives. Bundling speech with other information, such as text, images, video, and data, is an essential part of the multimedia communications revolution. By way of example, Table 2 lists of some of the multimedia applications that rely heavily on speech. The purpose of this list is to illustrate the types of multimedia applications that use speech and the requirements they impose on the speech coders they employ. At this point, we will only discuss the first two applications of Table 2. They serve as a useful introduction to speech coder attributes. This is followed by a section on speech-coding standards. At the end of this section, we return to the list of applications in Table 2 and discuss which speech coders would be used. Two important criteria in determining requirements for a speech coder for multimedia applications are whether the application contains a live conversation (versus those that use speech in a messaging or broadcast mode) and whether or not the speech content is being delivered over a real-time network connection. If there is a real-time network connection, the character of that network will also influence the selection of the speech coder.

1) Video Telephony/Conference and Business Conference with Data-Sharing Applications: In these applications, two or more parties engage in a multimedia phone call/conference. While a video call once required expensive special-purpose equipment, today it can be realized by adding a low-cost video camera to a multimedia PC. Marketing studies seem to indicate that consumers are more interested than businesses in making video calls. Businesses are more interested in bundling the voice call together with software-sharing applications, such as word-processing documents, spread sheets, presentations, etc. These multimedia phone calls could take place over an ISDN connection, over a POTS connection using a telephone bandwidth modem, over ATM or frame relay, or via corporate LAN's or the Internet. Different multiplexing protocols would be needed in each of these cases, and some significant differences exist in the requirements for the speech coders.

First, we look at the common characteristics of video telephony conferences and calls. Since all parties are able to speak at the same time, there is a conversational dynamic in the call. This means that relatively low (communication and signal-processing) delay is necessary to maintain that dynamic. As discussed in Section II-A, studies have shown that one-way end-to-end delay should be below 150 ms [7] in order to maintain normal conversational dynamics. If there are more than two parties involved in the call, a *conference bridge* is generally used in which all voice channels are decoded, summed, and then reencoded for transmission to their destination. The use of such a bridge can have the effect of doubling the processing delay and reducing the voice quality. This makes the delay budget much tighter. The specific network over which the multimedia call is transmitted also has an impact on the quality of the resulting call. Over the Internet, real-time connections with less than 150 ms of delay are unlikely for a variety of reasons of the type discussed in Section II-A. Over such connections, conversational dynamics will fall apart because of the excessive delay. With ATM or frame relay, there is a good chance that the goal of 150-ms end-to-end one-way delay can be achieved. With POTS or ISDN connections, network delay is generally not a factor. The POTS connection will necessarily mean that a low-bit-rate speech coder is required in order to operate reliably over a telephone bandwidth modem. For an ISDN connection, neither speech-coder bit rate nor delay is a major consideration in most cases. The range of speech coder bit rates for ATM, frame relay, and IP networks could vary from low (6.4 or 8 kb/s) to high (64 kb/s) depending on the network architecture and the overall capacity of the system.

In addition, on the Internet, it is quite likely that some encoded speech packets will be dropped. A speech coder must provide robust performance in spite of lost packets to be useful. We discuss this issue further in Section III-C4.

Another important characteristic of these applications is that they tend to be hands-free, in the sense that the participants will not be holding telephone handsets but instead

will be speaking into microphones that are either mounted near the PC or in some free-standing mode. Each user terminal will likely have an always-on open microphone. This means that the input speech will tend to be noisier than that of regular telephony. Hence, another requirement of the speech coder is that it should have relatively good performance for speech with noisy backgrounds. Another obvious implication is that the user terminals will require acoustic echo control.

Another important attribute of speech coders for teleconferencing is high computational complexity. With the availability of extremely powerful CPU chips (see Fig. 8), speech coders are now implemented directly on the host processor of multimedia terminals and workstations. This presents some computational challenges since the host processor must be used for other terminal and signal-processing functions as well.

In summary, these applications have helped define the four key speech-coder attributes that are almost always traded off against each other, namely, delay, bit rate, complexity, and speech-coder quality. These attributes will be discussed more formally in the next section.

2) Speech-Coder Attributes: In the preceding section, we reviewed some of the requirements that different types of multimedia applications can impose on speech coders. Implicitly, we described certain attributes that speech coders should have. In this section, we review speech-coding attributes more formally. Speech-coding attributes can be divided into four categories: bit rate, delay, complexity and quality. The applications engineer determines which attributes are the most important. It is possible to relax requirements for the less important attributes so that the more important requirements can be met.

Bit rate is the attribute that most often comes to mind first when discussing speech coders. The range of bit rates that have been standardized is from 2.4 kb/s for secure telephony to 64 kb/s for network applications. Table 3 shows a list of standardized speech coders. It includes their bit rates, delays measured in frame size and look-ahead requirements, and their complexity in MIPS [8]. Of primary interest here are the coders standardized by the ITU. The 64-kb/s G.711 pulse code modulation (PCM) coder is used in digital telephone networks and switches throughout the world. The 32-kb/s G.726 adaptive differential (AD)PCM coder is used for circuit multiplication equipment to effectively increase the capacity of undersea cable and satellite links. The 64/56/48-kb/s G.722 and 16-kb/s G.728 coders are used in video teleconferencing over ISDN or frame relay connections. The G.723.1 and G.729 coders have been standardized for low-bit-rate multimedia applications over telephony modems. The standards for digital cellular and secure telephony have been included in Table 3 for completeness.

The second key attribute of speech coders, the *delay*, can have a large impact on individual coder suitability for a particular application. As discussed above, speech coders for real-time conversations cannot have too much delay or they will quickly become unsuitable for network

Table 3 ITU, Cellular, and Secure Telephony Speech-Coding Standards

Standard	Bit rate	Framesize/ Look-ahead	Complexity
ITU Standards			
G.711 PCM	64 kb/s	0 / 0	0 MIPS
G.726 [G.721*, G.723*], G.727 ADPCM	16, 24, 32, 40 kb/s	0.125 ms / 0	2 MIPS
G.722 Wideband Coder	48, 56, 64 kb/s	0.125 / 1.5 ms	5 MIPS
G.728 LD-CELP	16 kb/s	0.625 ms / 0	30 MIPS
G.729 CS-ACELP	8 kb/s	10 / 5 ms	20 MIPS
G.723.1 MPC-MLQ	5.3 & 6.4 kb/s	30 / 7.5 ms	16 MIPS
G.729 CS-ACELP Annex A	8 kb/s	10 / 5 ms	11 MIPS
Cellular Standards			
RPE-LTP (GSM)	13 kb/s	20 ms / 0	5 MIPS
IS-54 VSELP (TIA)	7.95 kb/s	20 / 5 ms	15 MIPS
PDC VSELP (RCR Japan)	6.7 kb/s	20 / 5 ms	15 MIPS
IS-96 QCELP (TIA)	8.5/4/2/0.8 kb/s	20 / 5ms	15 MIPS
PDC PSI-CELP (RCR Japan)	3.45 kb/s	40 / 10 ms	40 MIPS
U.S. DoD Secure Telephony			
FS-1015 LPC-10E	2.4 kb/s	22.5 / 90 ms	20 MIPS
FS-1016 CELP	4.8 kb/s	30 / 7.5 ms	20 MIPS
MELP	2.4 kb/s	22.5 / 20 ms**	~40 MIPS

* G.721 and G.723 were merged into G.726 in 1990.

** In addition to look-ahead, there is an additional 3 msec delay for the postfilter.

applications. On the other hand, for multimedia storage applications, with only one-way transmission of speech, the coder can have virtually unlimited delay and still be suitable for the application. Psychologists who have studied conversational dynamics know that if the one-way delay of a conversation is greater than 300 ms, the conversation will become more like a half-duplex or a push-to-talk experience rather than an ordinary conversation. In contrast, if a speech or audio file is being downloaded to a client terminal, an overall delay of 300 ms or more before starting will be virtually imperceptible to the user. Thus, a conversation is an application that is the most sensitive to coder delay, while one involving speech storage is the least delay-sensitive application.

As seen in Table 3, the highest rate speech coders, such as G.711 PCM and G.726 ADPCM, have the lowest delay. To achieve higher degrees of compression, speech must be divided into blocks or frames and then encoded a frame at a time. For G.728, the frames are five samples (0.625 ms) long. For the first-generation cellular coders, the frames are 20 ms long. This does not account for the full delay, however. The components of the total system delay include the frame size, the look ahead, other algorithmic delay, multiplexing delay, processing delay for computation, and transmission delay. The algorithm used for the speech coder will determine the frame size and the look-ahead. Its complexity will have an impact on the processing delay. The network connection will determine the multiplexing and transmission delays.

In the past, speech coders were only implemented on DSP chips or on other special-purpose hardware. Recent multimedia speech coders, however, have been implemented on the host central processing unit (CPU) of personal

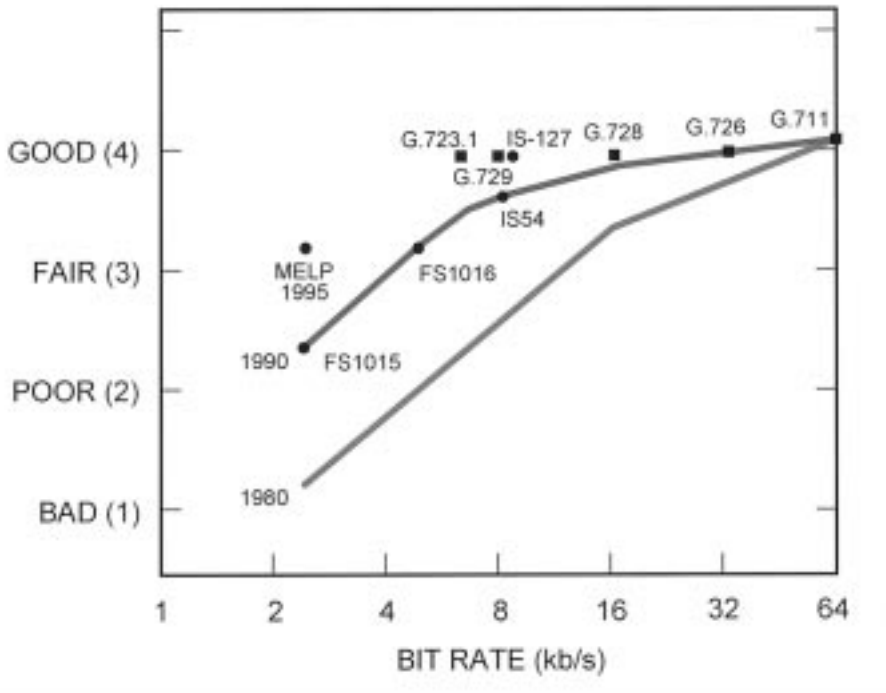

Fig. 11. Subjective quality of various speech coders versus bit rate.

computers and workstations. The measures of *complexity* for a DSP and a CPU are somewhat different due to the natures of these two systems. At the heart of complexity is the raw number of computational instructions required to implement the speech coder. DSP chips from different vendors have different architectures and consequently different efficiencies in implementing the same coder. The measure used to indicate the computational complexity is the number of instructions per second required for implementation. This is usually expressed in MIPS. The numbers given in Table 3 are DSP MIPS for each coder, and they range from zero to 30 MIPS for all the ITU standards-based coders.

Quality is the speech-coder attribute that has the most dimensions [9]. When speech coding was only used for secure telephony, quality was synonymous with intelligibility. The first requirement for secure speech communications was that the decoded speech be intelligible. Hence, the earliest speech coders that were used for security focused almost entirely on intelligibility of the coded speech. For network applications, the quality story was entirely different. High intelligibility of the coded speech was assumed, and the goal was to preserve the naturalness or so-called subjective quality of the original speech. The earliest telephone network speech coders operated on a sample-by-sample basis. Their quality was more or less directly related to the signal-to-noise ratio (SNR) they achieved in quantizing the speech samples. A high enough SNR for speech guaranteed a high SNR for other audio signals that might also be encountered, such as background noise or music. At the low bit rates used for secure telephony, however, the speech was coded based on a speech production model. This model was barely capable of modeling speech and could not handle music or any combination of speech plus some other signal. The result was that when the input speech was coded at low bit rates, the quality of the coded speech degraded significantly. Much of the research of the past decade has been on trying to make low-bit-rate speech coders work well on clean speech and make them more robust to extraneous noises and signals. This problem of robustness for noisy input speech has been observed for all speech coders that attempt to code speech at rates below 16 kb/s.

The "ideal" speech coder has a low bit rate, high perceived quality, low signal delay, and low complexity. No ideal coder as yet exists with all these attributes. Real coders make tradeoffs among these attributes, e.g., trading off higher quality for increased bit rate, increased delay, or increased complexity. To illustrate the current status of quality of telephone bandwidth coders, Fig. 11 shows a plot of speech quality [as measured subjectively in terms of mean opinion scores (MOS's)] for a range of telephone bandwidth coders spanning bit rates from 64 kb/s down to 2.4 kb/s. Also shown are curves of quality based on measurements made in 1980 and 1990.

The MOS subjective test of speech quality uses a five-point rating scale with the attributes:

5 excellent quality, no noticeable impairments;

4 good quality, only very slight impairments;

3 fair quality, noticeable but acceptable impairments;

2 poor quality, strong impairments;

1 bad quality, highly degraded speech.

It can be seen from Fig. 11 that telephone bandwidth coders maintain a uniformly high MOS score for bit rates ranging

from 64 kb/s down to about 8 kb/s but fall steadily for coder bit rates below 8 kb/s.

3) Speech-Coding Standards: Standards play a major role in the multimedia revolution because they provide interoperability between hardware and software provided by multiple vendors. Although, in practice, one could use a nonstandard coder, this would lead to a closed architecture system, which would discourage the widespread use of the system in which the coder was embedded. As a result, all of the major vendors of both terminals and software for multimedia communications have embraced the concept of standardization so that their various products will interoperate at a basic level. As the standards are written, there is provision for nonstandard coders to be used. At call setup, the terminals can exchange information on their capabilities. It is essential that at least one standard speech coder be mandatory in all terminals. This assures that voice communications can always be initiated and maintained. Additional optional speech coders can be included at the manufacturer's discretion.

In selecting a mandatory speech coder (also known as a default coder), the different standards bodies have selected from the list of ITU speech coders in Table 3. By selecting from the list, the standards bodies have the benefit of the extensive testing that the ITU has done to characterize each of these coders. Each of these coders is also very precisely specified, so that implementations by two different manufactures not only will interoperate but also will provide an assured level of quality known to match that observed in the ITU testing.

In this section, we review the classes of speech coders in Table 3, with comments on the use and appropriateness of these coders for various multimedia applications, such as those listed in Table 2. The coders can be broken down into particular classes that illustrate the current state of the art in speech coding.

a) Direct sample-by-sample quantization G.711: The G.711 PCM coder is designed for telephone bandwidth speech signals and does direct sample-by-sample quantization. Rather than using uniform quantization, the 8-b code word is similar to the representation of a floating-point number in a computer—one part represents the linear fraction, while the other part is related to the exponent. This quantization technique does not work well for wider bandwidth speech signals because the quantization noise becomes audible with higher bandwidths. For telephone bandwidth signals, G.711 PCM gives the most robust performance for any input signal since it is not a speech specific coder. It provides the lowest delay possible (a single sample) and provides the lowest complexity possible as well. On the negative side, G.711 is a high-rate coder at 64 kb/s, is very susceptible to bit errors, and has no recovery mechanism for lost packets. Thus, G.711 would be a poor choice to use over a packet or wireless network with an unreliable quality of service. G.711 is the default coder for ISDN video telephony or extensions to shared data applications. It needs network connections of 128 kb/s or greater, however, to be effectively integrated with a low-bit-rate video coder.

b) Wide-band speech coder G.722: The wide-band speech coder G.722 operates at 64, 56, or 48 kb/s. It is designed for transmitting 7-kHz bandwidth voice or music. The quality is not perfectly transparent, especially for music. This is because the upper band (4–7 kHz) is quantized using 2-b/sample ADPCM. Nevertheless, for teleconference-type applications, G.722 is greatly preferred to G.711 PCM because of the increased bandwidth. It is much less fatiguing to listen to for long periods of time. It does not have any frame- or packet-loss concealment strategy. Most of the delay in G.722 is due to a 24 tap quadrature mirror filter used in both the encoder and decoder in order to divide the speech into two bands. This causes a delay of 1.5 ms. A project is presently under way within the ITU to create and standardize a second wide-band speech coder operating at bit rates of 16, 24, and 32 kb/s.

c) Backward ADPCM G.726 and G.727: The speech coders G.726 and G.727 are backward ADPCM coders for telephone bandwidth speech. They operate on a sample-by-sample basis by predicting the value of the sample and then quantizing the difference between the actual value and the predicted value. A linear predictor is used based on the two previous output values and the six previous quantizer values. The predictor functions are updated on a sample-by-sample basis in a backward adaptive fashion. The levels of the quantizer are also updated in a backward adaptive fashion. Both coders can operate using 2, 3, 4, or 5 b/sample, corresponding to rates of 16, 24, 32, and 40 kb/s. The difference between these two coders is that G.727 uses embedded quantizers while G.726 uses individually optimized quantizers for each of the four bit rates. The principal rate for both coders is 32 kb/s. The performance at this rate is almost equivalent to that of G.711 PCM. G.726 was created for circuit multiplication applications. G.727 was created for packet circuit multiplication applications. The embedded quantizer feature allows the least significant bits to be dropped if there is network congestion. Not only can these coders perform robustly for speech but they also can pass most modem signals below 4800 b/s with their 32-kb/s rates and can pass modem signals at rates of up to 9600 b/s at 40-kb/s rates.

d) Linear prediction analysis-by-synthesis (LPAS) coders: By far, the most popular class of speech coders for bit rates between 4.8 and 16 kb/s are model-based coders that use an LPAS method. A linear prediction model of speech production is excited by an appropriate excitation signal in order to model the signal over time. The parameters of both the speech model and the excitation are estimated and updated at regular time intervals (e.g., every 20 ms) and used to control the speech model. In the following two sections, we distinguish between forward adaptive and backward adaptive LPAS coders.

e) Forward adaptive LPAS coders—8-kb/s G.729 and 6.3- and 5.3-kb/s G.723.1: In a forward adaptive analysis-by-synthesis coder, the prediction filter coefficients and gains are explicitly transmitted. To provide toll-quality performance, these two coders rely on a source model for

speech. The excitation signal, in the form of information on the pitch period of the speech, is transmitted as well. The linear predictive coding (LPC) filter is tenth order and is augmented by a pitch predictor. This provides a good model for a speech signal but is not an appropriate model for some noises or for most instrumental music. Thus, the performance of LPAS coders for noisy backgrounds and music is poorer than G.726 and G.727 coders.

G.723.1 provides toll-quality speech at 6.4 kb/s. A lower quality speech coder operating at 5.3 kb/s is also included. G.723.1 was designed with low-bit-rate video telephony in mind. For this application, the delay requirements are less stringent because the video coding delay is usually so much larger than that of speech. The G.723.1 coder has a 30-ms frame size and a 7.5-ms look ahead. When combined with processing delay to implement the coder, it is estimated that the coder would contribute 67.5 ms to the one-way delay. Additional delays result from the use of network and system buffers. G.729 was designed for low-delay applications, with a frame size of 10 ms and a look ahead of 5 ms. This yields a 25-ms contribution to end-to-end delay and a bit rate of 8 kb/s. G.729 comes in two versions—the original version is more complex than G.723.1, while the Annex A version is less complex than G.723.1. The two versions are compatible but their performance is somewhat different, the lower complexity version having slightly lower quality. Both coders include provision for dealing with frame erasures and packet-loss concealment, making them good choices for use with voice over the Internet. Their performance for random bit errors is poorer. They are not recommended for use on channels with random bit errors unless there is a channel coder to protect the most sensitive bits.

f) Backward adaptive LPAS coding—16-kb/s G.728 low-delay code book excitation linear prediction (LD-CELP): G.728 is in a class by itself, sort of a hybrid between the lower bit rate linear predictive analysis-by-synthesis coders (G.729 and G.723.1) and the backward ADPCM coders. G.728 is an LD-CELP coder. It operates on five samples at a time. It uses LPC analysis to create three different linear predictors. The first is a fiftieth-order prediction filter for the next sample values. The second is a tenth-order prediction filter that guides the quantizer step size. The third is a perceptual weighting filter that is used to select the excitation signal. CELP is a speech-coding paradigm in which the excitation signal is selected from a set of possible excitation signals via exhaustive search. Each possible excitation signal is evaluated to find the one that minimizes the mean square error in a perceptually weighted space. The perceptual weighting filter defines that space. While the lower rate speech coders use a forward adaptation scheme for the sample value prediction filter, LD-CELP uses a backward adaptive filter that is updated every 2.5 ms. This is possible because the output signal is close enough in value to the original signal that the output values can be used as the basis for LPC analysis. There are 1024 possible excitation vectors. These are further decomposed into four possible gains, two possible signs ($+$ or $-$), and 128 possible shape vectors.

G.728 is a suggested speech coder for low-bit-rate (56–128 kb/s) ISDN video telephony. Because of its backward adaptive nature, it is low delay but is also regarded as higher complexity. The fiftieth-order LPC analysis must be repeated at the decoder, and this is the single largest component of its "high complexity." It also features a sophisticated adaptive postfilter that enhances its performance. Although LPC analysis is widely associated with speech, the fiftieth-order LPC filter can capture the underlying redundancy in music. G.728 is considered equivalent in performance to 32-kb/s G.726 and G.727 ADPCM. Because of the adaptive postfilter, there is a frame- or packet-loss concealment strategy for G.728. This concealment strategy is not part of the present standard but is under consideration by the ITU. G.728 is remarkably robust to random bit errors, more so than any other speech coder. Moreover, the sensitivity to bit errors is roughly equal for all ten bits in a code word.

g) Parametric speech coders—2.4-kb/s mixed-excitation LPC (MELP): Parametric speech coders assume a generic speech model with a simplified excitation signal and thus are able to operate at the lowest bit rates. All of the speech coders discussed previously can be described as waveform following. Their output signals are similar in shape and phase to the input signal. Parametric speech coders do not have this property. They are based on an analysis-synthesis model for speech signals that can be represented using relatively few parameters. These parameters are extracted and quantized, usually on a regular basis from every 20–40 ms. At the receiver, the parameters are used to create a synthetic speech signal. Under ideal conditions, the synthetic signal sounds like the original speech. Under harsh enough background noise conditions, any parametric coder will fail because the input signal is not well modeled by the inherent speech model. The 2.4-kb/s MELP [10] was selected as the U.S. government's new 2400-b/s speech coder for secure telephony.

For multimedia applications, parametric coders are a good choice when there is a strong demand for low bit rate. For example, parametric coders are often used for single user games. This keeps down the storage requirements for speech. For the same reason, they also are a good choice for some types of multimedia messaging. They tend to be lower in absolute quality for all types of speech conditions and particularly noisy background conditions. This shortcoming can be overcome when the speech files can be carefully edited in advance. At the present time, most of the parametric coders being used in such applications are not standards. Rather, they are proprietary coders that have been adapted to work for such applications.

4) Selecting Speech Coders for Multimedia Applications: Having reviewed the different types of speech coders, we are now in a position to make recommendations for the different applications to multimedia communications listed in Table 2.

a) Video telephony/conference and distance learning/teaching: The selection of which coder to choose depends primarily on the network and bit rate being used

to transport the signals. For higher rate, more reliable networks such as ISDN, ATM, and frame relay, the logical choice is G.722 because it has the best quality. There is bit rate available to accommodate it at either the 56 or 64 kb/s rates. The 7-kHz bandwidth provided by G.722 makes the conferences or calls more pleasurable experiences for the participants. If the bandwidth is limited to the range of 56–128 kb/s, G.728 is a good choice because of its robust performance for many possible speech and audio inputs. If the bit rate is lower yet, such as that obtained using a telephone bandwidth modem, or if the network is less reliable, such as the Internet, then the best choice in speech coders is G.723.1.

b) Business conference with data sharing: In this application, the network is more likely to be a corporate intranet, an extranet, or the Internet. Depending on the quality of service of the network and the available bandwidth, the three best choices for speech coders are G.722, G.728, and G.729. The considerations are basically the same as those for video conferencing except that since video is not a part of this call, G.729 or G.729 Annex A is substituted for G.723.1. This reduces the delay in order to maintain conversational dynamics.

c) Single-user games: In this application, the voice is used primarily as a sound effect. It can be prerecorded and processed to give the highest possible fidelity. The game only contains the bit stream and the speech decoder. Delay is not a consideration. To keep the size of the game reasonable, it is preferable to use the lowest bit rate speech coder that is suitable. This may well be a parametric coder. It is not necessary to use a standard, since the voice files will not be exchanged with other games.

d) Multiuser games at remote locations: In this application, the game players can talk to each other. Now there are issues similar to those of a conversation. The dynamic of this conversation may well be quite different. This could change some of the implications for the delay. Typically, the connections are either a point-to-point modem link or via the Internet. The requirements associated with streaming apply if the Internet is used. It may also be the case that it is desirable to disguise the voices of the players. Parametric coders are quite adept at this, since they implicitly break down the voice to a few parameters. As in single-user games, very low bit rate is required. The coder should be low in complexity because the terminals must do both encoding and decoding.

e) Multimedia messaging: In this application, messages are sent that include speech, perhaps combined with other nonvoice information such as text, graphics, images, data, or video. This is an example of an asynchronous communication. Hence, delay is not an issue. Since the messages may be shared with a wide community, the speech coder used ought to be a commonly available standard. The network over which these messages are shared will determine the bit-rate limitations. For the most part, fidelity will not be an issue. Thus, coders such as G.729 or G.723.1 seem like good candidates. At the same time, low-bit-rate parametric coders would also make good candidates, provided that all parties agree to a *de facto* standard.

f) Voice annotated documents: Multimedia documents can be created that include speech as either annotations or an integral part of the document. Essentially, this could be treated as a type of messaging application where the document is always downloaded to the end user's terminal/workstation. In effect, this is much like the single-user game application. Selecting a standard is not necessary. To keep the storage size to a minimum, low bit rate is to be encouraged. Unlike the single-user game application, the voice input is probably going to be from an open microphone at a workstation. Low-bit-rate parametric coders may not perform as robustly for such signals. The selection of the coder will ultimately depend on the tradeoff among bit rate, complexity, and performance with open microphone signals.

C. CD-Quality Audio Coding for Multimedia Applications

A key aspect of multimedia systems is the ability to provide CD-quality audio over telecommunications networks. Since, as shown in Table 1, uncompressed CD audio requires 1.4 Mb/s for transmission, high-quality coding of CD audio is essential for its practical use in any multimedia application. The state of the art in CD audio coding has improved dramatically over the course of the last decade. This is due primarily to the relentless exploitation of known and well-understood properties of the human auditory system. Almost all modern CD audio coders use a quantitative model of the human auditory system to drive the signal quantization so that the resulting distortion is imperceptible. Those features of the audio signal that are determined not to be audible are discarded. Moreover, the amount of quantization noise that is inaudible can also be calculated. This combination of discarding inaudible features and quantizing the remaining features so that the quantization noise will be inaudible is known as perceptual coding. It has made its greatest impact to date in the field of audio coding and has been extended to speech, image, and video coding. In the next section, we discuss perceptual coding in greater detail, giving examples of how the analysis is carried out and how CD audio coding algorithms are actually structured. In the following section, we give a description of the audio coders that have been standardized to date. Then we describe some of the applications that digital audio coding has made possible.

1) Perceptual Audio Coding: Source coding removes *redundancy* in a signal by estimating a model of the source. Typically, the source coder uses the source model to increase the SNR or some other quality metric of the signal by the appropriate use of signal models and mathematical redundancies. Numerical coders use entropy-based methods to reduce the actual entropy, or bit rate, of the signal. As with source coders, the attempt is to remove the redundancy of the original signal. Perceptual coders use a model of the human perceptual apparatus to remove the parts of the signal that the human cannot perceive. Like most source coders, perceptual coding is a lossy coding method—the

Table 4 Critical Bands from Sharf

Band No.	Low	Center	Upper	Band No.	Low	Center	Upper
1	0	50	100	14	2000	2150	2320
2	100	150	200	15	2320	2500	2700
3	200	250	300	16	2700	2900	3150
4	300	350	400	17	3150	3400	3700
5	400	450	510	18	3700	4000	4400
6	510	570	630	19	4400	4800	5300
7	630	700	770	20	5300	5800	6400
8	770	840	920	21	6400	7000	7700
9	920	1000	1080	22	7700	8500	9500
10	1080	1170	1270	23	9500	10500	12000
11	1270	1370	1480	24	12000	13500	15500
12	1480	1600	1720	25	15500	19500	
13	1720	1850	2000				

output signal is not the same as the input signal. The imperceptible information removed by the perceptual coder is called the *irrelevancy.* In practice, most perceptual coders attempt to remove both irrelevancy and redundancy in order to make a coder that provides the lowest bit rate possible for a given quality. Ironically, most perceptual coders have lower SNR than source coders but have better subjective quality for an equivalent bit rate.

The human auditory system (HAS) has remarkable detection capability with a range of over 120 dB from very quiet to very loud sounds. However, louder sounds mask or hide weaker ones. Specifically, the HAS shows a limited detection ability when a stronger signal occurs close (in frequency) to a weaker signal such that a 30-dB SNR is *transparent* in one *critical band* (these two terms are defined below). In many situations, the weaker signal is imperceptible even under the most ideal listening conditions. In audio coding, we can think of the quantization noise as the weaker signal and the original signal as the stronger one. If we can confine the quantization noise to be weak enough so as to be imperceptible, the listener will not be able to hear it.

These concepts have been formalized in the theory of critical-band masking. The concept of *critical bandwidth* dates back over 60 years to experiments of H. Fletcher [11] (the term "critical bandwidth" was coined later). Simply stated, a critical band is a range of frequencies over which the masking SNR remains more or less constant. One of Fletcher's experiments illustrates the concept. A tone at 700 Hz can mask narrow-band noise of lesser energy, provided that it is within approximately ± 70 Hz. If the noise is outside the range 630–770 Hz, then it must have much less energy than the tone. Thus, the critical bandwidth is 140 Hz. Table 4 gives the critical-band frequencies [12]. Below 500 Hz, the critical bandwidths are all about 100 Hz. Above 500 Hz, they grow in bandwidth. If we appropriately warp the frequency scale, we can make the shape of the critical-band filters nearly invariant.

A second observation about masking is that noise and tones have different masking properties. If we denote B as the critical-band number, then the following empirical rules have been observed:

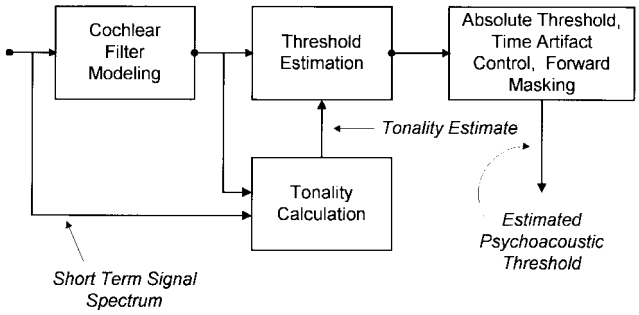

Fig. 12. Typical perceptual model for audio signal coding.

$$\text{Tone Masking Noise:} \quad E_N = E_T - (14.5 + B)\ \text{(dB)}$$
$$\text{Noise Masking Tone:} \quad E_T = E_N - K\ \text{(dB)}$$

where K has been assigned values in the range of 3–6 dB. In other words, for the tone masking noise case, if the energy of the noise (in decibels) is less than E_N, then the noise will be imperceptible. The problem is that speech and audio signals are neither pure tones nor pure noise but rather a mixture of both. The degree to which within a critical band a signal appears more or less tone-like (or noise-like) determines its masking properties.

A third observation is that loud sounds can mask portions of weaker sounds that follow them, and even some that precede them. This is due to the time constants of the auditory system. The implication is that we must be cognizant of the energy distribution in time as well as frequency when we attempt to code the signal. This masking behavior actually is a result of the cochlea filter bank and detector behavior that is manifested in the inner ear or cochlea. The mechanical mechanism in the human cochlea constitutes a filter bank. The response of the filter at any one position on the cochlea is called the cochlea filter for that point on the cochlea. A critical band is very close to the 3-dB bandwidth of that filter. The signals passing through this filter are those that are passed along to the auditory part of the brain. The tone in the above example excites the auditory nerve for 700 ± 70 Hz. The brain does not receive any signals corresponding to the noise. The implications of how the cochlea works explain many different auditory phenomena. We shall not attempt to discuss other phenomena of the human auditory system [13]. Exactly how the human auditory system works is still a topic of active research. For the purposes of those constructing perceptual audio coders, it is sufficient to be able to describe the phenomena in a mathematical sense. Then the various masking phenomena can be exploited to reduce the bit rate of the audio coder.

We now turn to the task of using what is known about perception to construct an audio coder. Fig. 12 is a block diagram of a typical psychoacoustic model of hearing. The input signal in this diagram is actually the short-term signal spectrum. The cochlea filter model computes the short-term

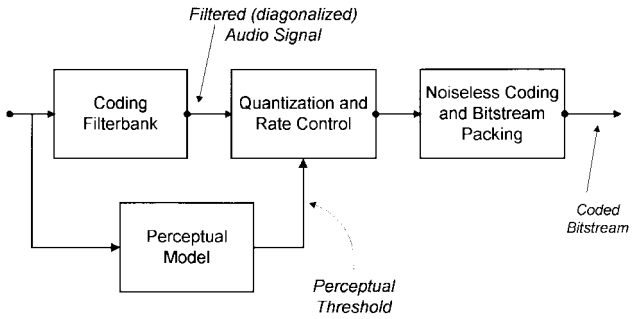

Fig. 13. Generic perceptual audio coder.

cochlea energy model, i.e., an indication of the energy along the cochlea. This information is used in two ways. The short-term signal spectrum and the cochlea energy are used to compute the tonality of the signal. Tonality is needed because tones mask noise better than noise masks tones. The cochlea energy plus the tonality information allow us to compute the threshold of audibility for the signal. This is a spectral estimate indicating what noise level will be audible as a function of frequency. As long as the quantization noise is less than the threshold of audibility at all frequencies, the quantization noise will be imperceptible.

Fig. 13 is a diagram for an entire perceptual audio coder [14]. The perceptual model from Fig. 12 is included in one of the boxes. However, this perceptual model also includes the conversion from the time-domain signal to a short-term signal spectrum. The coding filter bank is selected to give adequate resolution in both time and frequency for the input signal. In some audio coders, the time and frequency resolution can be changed by using different sized transforms according to the distribution of signal energy over time for a given block of the signal. This is commonly referred to as *window switching* because the window associated with the coding filter bank is switched from a high-frequency/low-time resolution window to one with low-frequency/high-time resolution. The output from the coding filter bank can be quantized based on the perceptual thresholds. Often, uniform quantization is used. The output code words are then further compressed using noiseless coding, most often a Huffman coder. Note that the quantization methodology employed here removed the irrelevancy and some redundancy. The use of the Huffman coder helps remove more redundancy as well. Because the exact bit rate of the final output cannot be predicted *a priori,* the perceptual thresholds can be adjusted either higher or lower to yield a different bit rate as needed to avoid buffer overflow or underflow. Of course, this also means that some quantization could be perceptible.

2) Audio-Coding Standards for Use in Multimedia Applications: In this section, we briefly describe some of the audio coders that have already been standardized for use in multimedia applications. The most important standards body in this domain is the ISO Motion Pictures Experts Group (ISO-MPEG) subgroup on high-quality audio [15]. (It may seem odd that a video and movies group should have responsibility for high-quality sound, but then who is better to make these standards, since sound movies and video are inherently multimedia signals?) ISO-MPEG has defined three sets of standards for high-quality audio and is already at work on a fourth. The original MPEG audio standard was created for monaural sound systems and had three layers, each providing greater amounts of compression. MPEG-2 was created to provide stereo and multichannel audio capability. The most recent standard is the MPEG advanced audio coder (MPEG-AAC,) which duplicates the performance of MPEG-2 at half the bit rate. In addition to these three standards, we also discuss the Dolby AC-3 coding system that is used in motion pictures and has been selected for HDTV in North America.

a) The original MPEG coder—Layers 1, 2, and 3: The original MPEG coder is sometimes now referred to as MPEG-1. The layers are successively more complex, but each provides a greater amount of compression. If we refer to Fig. 13, the coding filter bank has 32 equally spaced bands, each 750 Hz wide. The filters are 511 tap polyphase filters. The perceptual model is based on a 512-point fast Fourier transform (FFT) analysis. There are 15 possible quantizers for each band. Each block consists of 12 samples per band (8 ms). A dynamic bit allocation is computed for every block to allocate the bits among the bands. At 384 kb/s per channel, the Layer 1 coder is deemed to be transparent, meaning that the quantization noise is imperceptible for nearly all signals. Note that stereo would require 768 kb/s using two Layer 1 coders. The Layer 2 coder uses an improved perceptual model based on a 1024-point FFT analysis. It also has finer resolution quantizers and removes redundancies in coding some of the side information. As a result, it is deemed to be transparent at a rate of 256 kb/s per channel. The Layer 3 coder computes a modified discrete cosine transform (MDCT) on the samples of each band. There is a window-switching mechanism such that during stationary regions, the MDCT size is 18, while for blocks with unequal energy distribution in time, three MDCT's of size six are used. A nonuniform quantizer is used for the frequency components. Further rate reduction is accomplished by using Huffman coding on this bit stream (this removes further redundancy). The transparent bit rate for monaural signals for Layer 3 coding is 96 kb/s.

To provide some stereo capability, joint stereo coding can be used with any of the layers. In the higher frequency bands, there are separate gains for each channel, but after normalization by the gain factors, the left and right channel signals are summed and coded jointly. For the Layer 2 coder, joint stereo coding saves about 25% of the bit rate. Instead of a rate of 512 kb/s using two independent Layer 2 coders, joint stereo coding reduces the rate for achieving transparency to 384 kb/s.

b) MPEG-2 coder: The purpose of MPEG-2 was to provide theater-style surround-sound capabilities. In addition to the left and right channels, there is a center channel in the front and left and right channels in the rear or sides, denoted as the surround channels. There are actually five

different modes of operation corresponding to monaural, stereo, three channel (left, right, and center), four channel (left, right, center, and rear surround), and the five channel described above.

A second goal of MPEG-2 was to achieve compatibility with MPEG-1. There are two kinds of compatibility. Forward compatibility means that MPEG-2 decoders can decode MPEG-1 bit streams. Backward compatibility means that MPEG-1 decoders can decode a portion of the MPEG-2 bit stream. This goal was achieved by 1) using MPEG-1 encoders and decoders as the component parts for MPEG-2 and 2) defining the MPEG-2 bit-stream syntax to be a composite of the MPEG-1 bit stream followed by a portion only relevant to MPEG-2, i.e., coding the other channels. A matrix is used to mix the five component signals into five composite channels. The first two of these are coded by an MPEG-1 joint stereo coder. The other three are coded by the three-channel MPEG-2 extension coder. Full five-channel surround sound with MPEG-2 was judged to be transparent at a rate of 640 kb/s.

c) MPEG-AAC coder: MPEG-AAC was created because the technology in MPEG-2 was judged to be inferior to that of other commercially available systems. In a 1994 test, both the Dolby AC-3 system and the AT&T perceptual audio coder (PAC) were judged far superior to MPEG-2 in performance. Because the differences were so dramatic, ISO-MPEG authorized the audio-coding experts to create a nonbackwards-compatible coder, which was dubbed MPEG2-NBC. When the coder was almost finalized, the name was changed to MPEG-AAC. In the critical listening tests of the final version, MPEG-AAC at 320 kb/s was judged to be equivalent to MPEG-2 at 640 kb/s for five-channel surround-sound listening. While the coder was tested primarily for five channels and 320 kb/s, it can operate with other numbers of channels and bit rates. The number of channels can range from one to 48, and the composite bit rate starts at 8 kb/s per channel (kb/s/ch) and goes up from there to as much as 192 kb/s/ch. The sampling rate can be as low as 8 kHz and as high as 96 kHz per channel. The structure of the MPEG-AAC coder exactly mirrors that of Fig. 13.

d) Dolby AC-3 coder: The Dolby AC-3 coder was selected for the U.S. HDTV system and the digital video disc (DVD) system and is used in movie theaters as part of what is called the Dolby digital film system. For commercial cinema, the bit rate used is 64–96 kb/s/channel for five full channels plus a very low, very loud woofer (low-frequency) channel.

3) The New World of Audio: The combination of the Internet, World Wide Web browsers, streaming, and high-quality, low-bit-rate audio coders has created a whole new means for audio distribution. In fact, once there is greater availability of higher access rate channels, this distribution means will be applicable to all multimedia signals. In this section, we describe the combination of technologies that could be utilized to create the mainstream audio distribution channel of the future. There are hurdles to its becoming an accepted distribution channel. Some are not technical but revolve around economics, copyrights, electronic commerce, etc. This section is a brief description of a complex topic.

First, we describe several future audio distribution possibilities. All begin with the assumption that access to CD-quality audio is available over the Internet. In today's terminology, we assume that the music distributor has a Web page that the customer can reach with any browser. The Web then becomes the most comprehensive music store ever conceived. Any recording can be digitized and sold on the Web. The major labels will each have their own Web sites, as can any budding musician.

The scenario for electronic commerce with digital audio is as follows. The customer must first decide whether to listen or just browse. The customer also must decide whether to select music by artist, composer, style, or any other criteria, since every title can be cross-indexed in multiple categories. Just as there are search engines to help the user find all kinds of information on the Web, such programs will help the customer find whatever music they desire.

The customer must also decide how the digital music is delivered. The choices include buying a selection or listening to it just once. For the listening option, the customer chooses between listening immediately or downloading the selection and listening to it at a later time. Another potential customer choice would be to pay less for the music selection by accepting a digitized version with lower fidelity. The interesting part of this scenario is that the technology is already available to do all of these things now. Streaming technology (see the next section) makes real-time listening possible. To obtain full CD quality in real time requires a network connection with at least 100-kb/s capability or more, e.g., a LAN or an ISDN telephony connection. Unfortunately, over today's 28.8-kb/s telephone modems, CD-quality audio is not attainable. At telephone modem rates, however, audio with near FM quality can be reliably delivered, and this is the basis for providing lower quality audio at a lower price. There are numerous sites on the Web already where one can download "ripped" CD's. These are either selections or entire CD's that have been compressed and stored for others to copy. Typical compression rates are 112 or 128 kb/s using the MPEG Layer 3 audio coder described in the previous section. With high-speed LAN or ISDN access, these files can be downloaded faster than real time. With a standard 28.8-kb/s telephony modem, downloading these files takes considerably longer than real time.

Naturally, the above scenario brings up the next important issue. From the perspective of the copyright owners, a key question is how the audio material is protected so that it is not used to make an unlimited number of illicit copies. This same kind of issue held up the introduction of digital audio tape in the United States for years. The issue is how to address the concerns of the copyright holders and still enable the consumer to use the audio in the same way that they use it when they purchase a CD or cassette tape today—it may be for themselves or it may be for a gift; both

are recognized as perfectly legitimate uses. Hence, a key issue is if the bit stream is encrypted just for the purchaser, how can that person give it away to someone else? Software systems are being created to address these concerns [16]. In addition, the entire transaction over the Internet also needs to be made secure so that both the vendor and the consumer are certain that their account information and the bit-stream file are not stolen by some third party [17].

A third issue is how customers would like to store the selections they purchased. The first consideration is the system on which the recording is played. People do not want to carry a PC with them to listen to music, and they really do not have to. A relatively low-cost microprocessor or DSP chip can be used to decode digital audio bit streams. The compressed streams can be stored on a memory device with the form factor that works best. For instance, they could be temporarily stored on a flash random-access memory card and played in a portable device. Our colleagues have built a prototype, namely, a portable digital audio player with no moving parts (other than the buttons that control it). Such low-cost players could be used in home entertainment systems, in portable players, or in automobile audio systems.

If the copyright problems and other security issues can be overcome in ways that are satisfactory to both the copyright owners and the consumer, digital audio distribution via the Web could revolutionize the audio industry. In the current system, the big music companies only have general demographic information as to who are their customers. Over the Web, they can sell directly to their customers. In the process, they can gain valuable marketing information about their customers, helping the companies to target them specifically for future sales. The inventory problem goes away. Albums can be kept on-line for decades if so desired. Perhaps the top 1000 albums are given higher access speed servers to keep up with demand, but even obscure albums can be kept on-line. As this mode of commerce is developed for audio, it will set a precedent for distributing other multimedia information as well.

4) Streaming for Speech and Audio [18]: Streaming refers to the transmission of multimedia signals for real-time delivery without waiting for the entire file to arrive at the user terminal. Streaming can be either *narrowcast* (from the server to just one client) or *broadcast* (one transmission stream to multiple clients). The key point is that the real-time information is flowing solely from one source in both cases. There are four main elements to a streaming system:

1) the compressed (coded) information content, e.g., audio, video, speech, multimedia data, etc.;
2) the content, which is stored on a server;
3) the server, which is connected to the Internet and/or possibly other networks (POTS, ISDN, ATM, frame relay);
4) the clients.

Each of these elements can cause impairments. There are well-established methods for minimizing the degradations to the signal that result from these impairments. In this section, we discuss these component parts as well as the improvements that can be made.

To set the scenario, assume that a user has requested a real-time audio or video stream over the Internet. Further, we assume that the client is not directly connected to the stream on the Internet but instead accesses it through an ISP. The access could be through a modem on a POTS line, via ISDN, via a corporate LAN running IP, or could even include ATM or frame relay in the access link. Although the backbone of the Internet is a high-speed data network, there are a number of potential bottlenecks within the path from the server that streams the data to the final client that receives the real-time stream. For example, individual Web sites may have heavy traffic; the ISP may have heavy traffic; even a corporate LAN may have heavy traffic. Heavy traffic causes congestion and results in variable delays and possibly dropping packets. Thus, the two manifestations of network congestion on the packet stream that represents the real-time signal are highly variable delays and lost packets. The degree to which these two problems occur determines the quality of service that can be provided. The solutions to these problems must focus on these two issues—namely, delayed and lost packets.

One potential way of addressing these problems is via the compression scheme. Surprisingly, using compression to represent the real-time signal provides an excellent mechanism for dealing with lost packets. Since the signal has been analyzed and reduced to its component parts as part of the compression scheme, this often makes handling lost and delayed packets practical since parts of the lost signal are often highly predictable. Although the coding algorithms attempt to remove all redundancy in the signal, some redundancy still remains, even for the most advanced coders. We can also use concepts of statistical predictability to extrapolate some of the component parts for a missing packet. Hence, for both speech and audio, the best extrapolation policy for handling lost and delayed packets seems to be to assume stationarity of the missing signal. The degree to which this assumption is valid makes the strategy more or less viable in practice. If the signal is in a transitional state, holding its statistical properties constant will cause an obvious and highly perceptual degradation. However, shutting off the signal (i.e., playing silence in place of the signal) would lead to an even more obvious degradation most of the time. There is also the problem that lost packets cause the decoder state to lose synchronization with the encoder state. Generally, forward adaptive coders can resynchronize the encoder and decoder faster than backward adaptive coders. Hence, forward adaptive coders are preferred for highly congested data networks with streaming speech signals.

To understand the effects of extrapolation on streaming systems with lost and delayed packets, consider the processing used for LPAS speech coders (see Section III-B3). The usual assumption is that the pitch and LPC information of a speech signal will stay fairly constant during the time interval when the packets are lost or delayed. If this

interval is short enough (on the order of 20–30 ms), the extrapolation strategy will work quite well. For such short intervals, for over half the missing or lost frames, the resulting degradation is slight. If the interval of lost or delayed packets is longer in time (on the order of 40–90 ms), this strategy will not work well and will lead to excessive degradation of the speech. When the duration of lost or delayed packets exceeds 90 ms, the best thing to do is shut off the speech coder, since speech is almost never stationary for 90 ms or longer. The alternative of keeping the coder "turned on" during this interval causes a long and annoying "beep" if a periodic voiced sound is extended in time for that long. In general, these rules and observations are heuristic—if we could predict speech well for intervals of up to 90 ms, we could achieve much greater degrees of speech compression.

Conversely, if the speech or audio was uncoded or uncompressed, e.g., raw PCM format, an analysis of the immediate past material could be used to determine the amount of prediction that could be applied during periods of lost or delayed packets. For compressed material, as shown above, that information already exists.

The server's function in a streaming transaction is to transmit the information (the coded speech or audio) at an average rate designed to maintain real-time decoding of the compressed material. If the server is working too slowly (i.e., heavily overloaded), the real-time signal received by the client will have gaps caused by the decoder running out of bit stream to decode. If the server is transmitting too quickly (i.e., underloaded conditions), the bit stream will build up in the buffers of the decoder, eventually overfilling them and causing a loss of signal because of the buffer overflow. The server must serve multiple clients and must respond to changes in the network due to variable traffic and congestion. Thus, if the server is forced (requested) to reduce the amount of traffic it is generating on the network, it will be expected to do so. If it has too many clients, this can also cause interruptions in the regular transmission of packets. In general, just because it is serving many clients, the server will *not* transmit packets at regular intervals—there will necessarily be some jitter in its transmission instants. Additionally, it will be more efficient to transmit packets of larger size.

Obviously, the client must take some action to deal with the variation in time of receipt of packets due to the variable delay of packets. Most commonly, the delay is accounted for by creating a buffer at the client to smooth over the variations in delay. For information retrieval, this is generally acceptable. It means that the start of playback will be delayed in proportion to the size of the buffer, but for audio distribution, the extra delay will probably not be noticed.

The transmitted streaming information is now sent over the Internet and onto the user's network. As discussed above, the principal problem is congestion at a few nodes or over a few links. The congestion results in both variable delays and lost packets. If the rate of packet loss is too high, then real-time streaming is just not viable. The extrapolation techniques discussed above may be adequate to cover up losses of a few percent but will not suffice when the loss rate is in the tens of percent range. Similarly, if the delay variation is too great, then the longest delayed packets will be treated as lost packets. To the extent that this occurs, the quality of service may become unacceptable.

A technique that can be used with speech and also to some extent with audio is to change the buffer size to match the network delay characteristics. This method can also be used to account for any long-term differences in the clock rates of the server and the client.

Last, there are things that can be done with the client. As indicated above, the client software needs to be adaptive to the network so that it can respond to missing packets. The sampling rate clocks of the server and the client may not match. A technique to deal with this is to insert or delete parts of the packet as necessary. For speech, this is relatively easy to do. A pitch period can be repeated or deleted or silence interval durations can be adjusted. For general audio, the repeating or deleting techniques need to be slightly more sophisticated but are still possible. This technique can also be used to change the buffer size dynamically to match the current delay characteristics of the network.

The case of holding a real-time conversation over the Internet, so-called voice over (Vo)IP, is somewhat different than streaming. Obviously, there is no server in the sense described above. Instead, the system is composed of two or more clients talking to each other. Every client is capable of sending and receiving at least one voice stream. The protocol that has been selected, ITU-T Recommendation H.323, has a very rich set of capabilities. The voice channel is treated as one logical channel in the bit stream, but many more logical channels can be opened to carry image, video, audio, data, or even additional voice channels. The most significant difference between a conversation and streaming is the importance of delay. A delay of up to one second is not important for streaming as long as all parts of the multimedia signal stay synchronized thereafter. For a conversation, a delay of one second would effectively change it to a half-duplex or push-to-talk system. People cannot maintain conversational dynamics with such long delays. The Internet is one source of delay, but even the client hardware and software can often be a source of significant delay. The typical sound cards used in multimedia personal computers have buffers on both the input and output that cause delays. System software can also cause delays. Of the many degradations that VoIP users have observed, delay is the one most often cited first and is the most difficult to eliminate.

D. Image-Coding Overview

Image coding[4] generally involves compressing and coding a wide range of still images, including so-called bilevel or fax images, photographs (continuous-tone color or mono-

[4]We, along with others, use the term *image* for still pictures and *video* for motion pictures.

chrome images), and document images containing text, handwriting, graphics, and photographs. There are a number of important multimedia applications that rely heavily on image coding for both the consumer and business customer. These include the following.

- Slide-show graphics for applications such as ordering from catalogs, home shopping, real-estate viewing, etc. For this class of applications, it is essential that the images be presented at a variety of resolutions, including low resolution for fast browsing and searching (with a large number of individual, small-sized images displayed on a single page) and high resolution for detailed inspection. Ultimately, this type of application will demand a third mode of image display, namely, a 3-D model that allows the user to view the image from different frames of reference. This mode will clearly have great importance for viewing physical models such as furniture displays, house plans, etc.
- Creation, display, editing, access, and browsing of banking images (in the form of electronic checks) and forms (such as insurance forms, medical forms, etc.). For this application, it is essential that the user be able to interact with the system that creates and displays the images and that multimedia attachments be accessible easily by the user. For example, for insurance forms for accident claims, photographs of the accident scene are often attached to the form and should be accessible with the form.
- Medical applications where sophisticated high-resolution images need to be stored, indexed, accessed, and transmitted from site to site on demand. Examples of such images include medical X-rays, CAT scans, EEG and EKG scans, etc.

1) Basic Principles of Image Coding [19]: Unlike speech signals, which can take advantage of a well-understood and highly accurate physiological model of signal production, image and video signals have no such model on which to rely. As such, in order to compress and code image and video signals, it is essential to take advantage of any observable redundancy in the signal. The obvious forms of signal redundancy in most image and video signals include spatial redundancy and temporal redundancy.

Spatial redundancy takes a variety of different forms in an image, including correlations in the background image (e.g., a repeated pattern in a background wallpaper of a scene), correlations across an image (repeated occurrences of base shapes, colors, patterns, etc.), and spatial correlations that occur in the spectral domain. A variety of techniques have been devised to compensate for spatial redundancy in image and video sequences, and some of these will be discussed later in this section.

Temporal redundancy takes two different forms in video sequences. The most obvious form is redundancy from repeated objects in consecutive frames of a video sequence. Such objects can remain, they can move horizontally, vertically, or any combination of directions (translation movement), they can fade in and out (camera pans and fades), and they can disappear from the image as they move out of view. A variety of techniques have been devised to compensate for temporal redundancy in video sequences, and some of these will be discussed later in Section III-E.

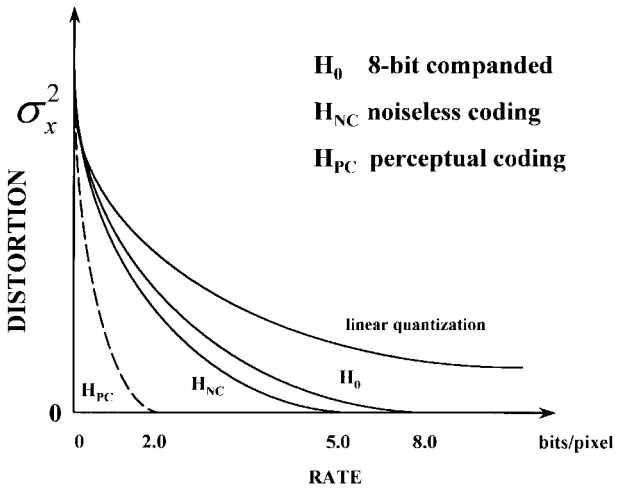

Fig. 14. Curves of perceptual entropy (as measured by image distortion) versus the coding rate (as measured in bits/sample). (From N. S. Jayant *et al.,* "Signal compression based on models of human perception," *Proc. IEEE,* vol. 81, pp. 1385–1422, Oct. 1993.)

The second basic principle of image coding is to take advantage of the human visual system, which is used to view the compressed and coded image and video sequences in much the same ways that we take advantage of the human hearing system for listening to speech and audio signals. Through various psychophysical testing experiments, we have learned a great deal about perception of images and video, and we have learned several ways to exploit the human's ability essentially to pay no attention to various types of image distortion. By understanding the perceptual masking properties of the human visual system, and their insensitivity to various types of distortion as a function of image intensity, texture, and motion, we can develop a profile of the signal levels that provide just noticeable distortion (JND) in the image and video signals. By creating this JND profile for each image to be coded, it is possible to create image quantization schemes that hide the quantization noise under the JND profile and thereby make the distortion become perceptually invisible.

To illustrate the potential coding gains that can be achieved using a perceptual coding distortion measure, consider the set of curves of perceptual entropy (distortion) of a black-and-white still image versus the image-coding rate measured in bits/sample (or equivalently bits/pixel), as shown in Fig. 14. The upper curve (simple linear quantization) in this figure shows that in theory, it would take more than 8 b/pixel to achieve low (essentially zero) distortion. By using proper companding methods, this zero distortion point can be reached with just 8 b/pixel in practice, as shown by the second curve in Fig. 14. By taking advantage of modern noiseless coding methods, (e.g., Huffman coding or arithmetic coding methods), the noiseless coding (NC) threshold can be reduced by a factor

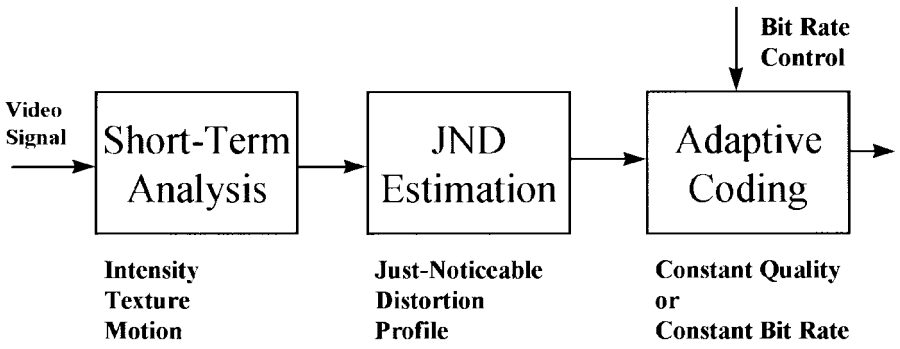

Fig. 15. Block diagram of a generic image-coding algorithm based on perceptual coding principles. (From N. S. Jayant *et al.*, "Signal compression based on models of human perception," *Proc. IEEE*, vol. 81, pp. 1385–1422, Oct. 1993.)

of about 2 to 1, down to 5 b/pixel, as shown by the third curve in Fig. 14. Last, by exploiting the perceptual model discussed above, the "perceptually lossless" coding threshold can be reduced by another factor of about three, down to around 2 b/pixel (or below depending on the image), as shown by the dotted curve in Fig. 14.

Based on the concepts discussed above, a generic image-coding algorithm assumes the form shown in Fig. 15. The first step in the process is to perform a short-term (short-range) spectral analysis of the input image or video signal in order to determine a profile of image intensity, image texture, and image motion. This short-term profile is then fed into a JND estimation box, which converts the measured image characteristics into a JND profile, which is then used as the input to an adaptive coding box, which provides either a constant-bit-rate (variable-quality) output or a constant-quality (variable-bit-rate) output signal.

By way of example, Figs. 16 and 17 show examples of a black-and-white (monochrome) still image (LENA) coded at rates of 8, 0.5, 0.33, and 0.25 b/sample and a color image (FROG) coded at rates of 24, 1, 0.5, and 0.25 b/sample. Although there are small differences in the images at the lower rates, most of the image characteristics are well preserved at all the coding rates.

2) Coding of Bilevel Fax Images [20]: The concepts behind fax coding of bilevel images have been well understood for more than 100 years. Until a set of standards was created and became well established, however, fax machines were primarily an office curiosity that were restricted to a few environments that could afford the costs of proprietary methods that could only communicate with like proprietary machines. Eventually, the industry realized that standards-based fax machines were the only way in which widespread acceptance and use of fax would occur, and a set of analog (Group 1 and Group 2) and digital standards [Group 3 (G3) and Group 4 (G4)] were created and widely used. In this section, we consider the characteristics of G3 and G4 fax [21], along with the more recent Joint Bilevel Image Group (JBIG)-1 standard and the newly proposed JBIG-2 standard.

To appreciate the need for compression of fax documents, consider the uncompressed bit rate of a scanned page (8 1/2 by 11 inches) at both 100 and 200 dpi. At 100 dpi, the single page requires 935 000 bits for transmission, and at 200 dpi, the single page requires 3 740 000 bits for transmission. Since most of the information on the scanned page is highly correlated across scan lines (as well as between scan lines), and since the scanning process generally proceeds sequentially from top to bottom (a line at a time), the digital-coding standards for fax process the document image line by line (or pairs of lines at a time) in a left-to-right fashion. For G3 fax, the algorithm emphasizes speed and simplicity, namely, performing a one-dimensional run-length coding of the 1728 pixels on each line, with the expedient of providing clever codes for end of line, end of page, and for regular synchronization between the encoder and decoder.

Fig. 16. Monochrome image coded at 8, 0.5, 0.33, and 0.25 b/sample. (The 8-b/sample coded image is at the upper left, the 0.5-b/sample coded image is at the upper right, the 0.33-b/sample coded image is at the lower left, and the 0.25-b/sample coded image is at the lower right.) (From N. S. Jayant *et al.*, "Image compression based on models of human vision," in *Handbook of Visual Communications*, H.-M. Hang and J. W. Woods, Eds. New York: Academic, 1995.)

Fig. 17. Color image coded at 24, 1, 0.5, and 0.25 b/sample. (The 24-b/sample coded image is at the upper left, the 1-b/sample coded image is at the upper right, the 0.5-b/sample coded image is at the lower left, and the 0.25-b/sample coded image is at the lower right.) (From N. S. Jayant *et al.*, "Image compression based on models of human vision," in *Handbook of Visual Communications,* H.-M. Hang and J. W. Woods, Eds. New York: Academic, 1995.)

The resulting G3 standard provides, on average, a 20-to-1 compression on simple text documents.

The G4 fax standard provides an improvement over G3 fax by using a two-dimensional (2-D) coding scheme to take advantage of vertical spatial redundancy as well as the horizontal spatial redundancy used in G3 fax coding. In particular, the G4 algorithm uses the previous scan line as a reference when coding the current scan line. When the vertical correlation falls below a threshold, G4 encoding becomes identical to G3 encoding. Otherwise, the G4 encoding codes the scan line in a vertical mode (coding based on previous scan line), a horizontal mode (locally along the scan line), or a pass mode (which essentially defers the encoding decision until more of the scan line is examined). The simple expedient of allowing the encoding to be based on the previous scan line increases the compression that is obtained on simple text documents, on average, to 25 to 1, a 25% improvement over G3 encoding.

G4 fax has proven adequate for text-based documents but does not provide good compression or quality for documents with handwritten text or continuous-tone images. As a consequence, a new set of fax standards was created

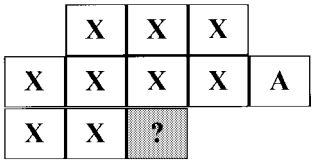

Fig. 18. The JBIG-1 sequential template, where "?" marks the pixel to be coded based on the previously coded pixels marked "X" and an adaptive pixel marked "A."

in the late 1980's, including the JBIG-1 standard [22], and work has begun on the more recent JBIG-2 standard. The key idea here is that for binary halftone images (i.e., continuous-tone images that are converted to dot patterns, as in newspapers), neither G3 nor G4 fax coding is adequate since image pixel predictability needs a significantly larger region of support for prediction than that needed for text images. As such, the JBIG-1 standard uses an arithmetic coder that is a dynamically adaptive binary coder that adapts to the statistics for each pixel context. There are two prediction modes that can be used for encoding. The first is a sequential mode in which the pixel to be coded is predicted based on nine adjacent and previously coded pixels plus one adaptive pixel that can be spatially separated from the others, as shown in Fig. 18. Since each previously

encoded pixel is a single bit, the previous pixels used in the sequential coding mode form a 10-b context index that is used to encode the current pixel arithmetically.

The second coding mode of JBIG-1 is a progressive mode that provides for successive resolution increases in successive encodings. This is useful in browsing applications, where a low-resolution image can be received fairly quickly, with higher resolution arriving later if the user wishes to wait.

The key behind JBIG-1 coding is that binary halftone images have statistical properties that are very different from binary text and therefore need a significantly different coding algorithm to provide high-quality encoding at significant compression rates. The JBIG-1 standard provides compression rates that are comparable to G4 fax for text sequences, and an improvement in compression by a factor of up to 8-to-1 is achieved for binary halftone images.

Although JBIG-1 compression works quite well, it has become clear over the past few years that there exists a need to provide optimal compression capabilities for both lossless and lossy compression of arbitrary scanned images (containing both text and halftone images) with scanning rates of 100–800 dpi. This need was the basis for the JBIG-2 method, which is being proposed as a standard for bilevel document coding. The key to the compression method is called "soft pattern matching" [23], [24], which is a method for making use of the information in previously encountered characters without risking the introduction of character-substitution errors that are inherent in the use of OCR methods [25].

The basic ideas of the JBIG-2 standard are as follows [26].

- The basic image is first segmented into individual marks (connected components of black pixels).
- The resulting set of marks is partitioned into equivalence classes, with each class ideally containing all occurrences of a single letter, digit, or punctuation symbol.
- The image is then coded by coding a representative *token* mark from each class, the position of each mark (relative to the position of the previous mark), the index of the matching class, and, last, the resulting error signal between each mark and its class token.
- The classes and the representative tokens are adaptively updated as the marks in the image are determined and coded.
- Each class token is compressed using a statistically based, arithmetic coding model that can code classes independently of each other.

The key novelty with JBIG-2 coding is the solution to the problem of substitution errors in which an imperfectly scanned symbol (due to noise, irregularities in scanning, etc.) is improperly matched and treated as a totally different symbol. Typical examples of this type occur frequently in OCR representations of scanned documents, where symbols like "o" are often represented as "c" when a complete loop is not obtained in the scanned document or a "t" is changed to a "l" when the upper cross in the "t" is not detected properly. By coding the bitmap of each mark, rather than simply sending the matched class index, the JBIG-2 method is robust to small errors in the matching of the marks to class tokens. Furthermore, in the case when a good match is not found for the current mark, that mark becomes a token for a new class. This new token is then coded using JBIG-1 with a fixed template of previous pixels around the current mark. By doing a small amount of preprocessing, such as elimination of very small marks that represent noise introduced in the scanning process, or smoothing of marks before compression, the JBIG-2 method can be made highly robust to small distortions of the scanning process used to create the bilevel input image.

The JBIG-2 method has proven itself to be about 20% more efficient that the JBIG-1 standard for lossless compression of bilevel images. By running the algorithm in a controlled lossy mode (by preprocessing and decreasing the threshold on an acceptable match to an existing mark), the JBIG-2 method provides compression ratios about two to four times that of the JBIG-1 method for a wide range of documents with various combinations of text and continuous-tone images. We will also see later in this paper that the JBIG-2 method forms the basis for a highly flexible document-compression method that forms the basis for the CYBRARY digital-library project.

3) Coding of Continuous Images—Joint Photographic Experts Group (JPEG) Methods [27]: In this section, we discuss standards that have been created for compressing and coding continuous-tone still images—both grayscale (monochrome) and color images, of any size and any sampling rate. We assume that the uncompressed images are available in a digital format, with a known pixel count in each dimension [e.g., the rates shown in Table 1(b)] and an assumed quantization of 8 b/pixel for grayscale images, and 24 b/pixel for color images.

The standard algorithm for compression of still images is called the JPEG algorithm [28]. It is of reasonably low computational complexity, is capable of producing compressed images of high quality, and can provide both lossless and lossy compression of arbitrarily sized grayscale and color images [29]. A block diagram of the JPEG encoder and decoder is shown in Fig. 19. The image to be compressed is first converted into a series of 8-by-8 (pixel) blocks, which are then processed in a raster scan sequence from left to right and from top to bottom. Each such 8×8 block of pixels is first spectrally analyzed using a forward DCT algorithm, and the resulting DCT coefficients are scalar quantized based on a psychophysically based table of quantization levels. Separate quantization tables are used for the luminance component (the image intensity) and the chrominance component (the color). The entries in the quantization tables are based on eye masking experiments and are essentially an approximation to the best estimate of levels that provide just-noticeable distortion in the image. The 8×8 blocks are then processed in a zigzag order following quantization. An entropy encoder

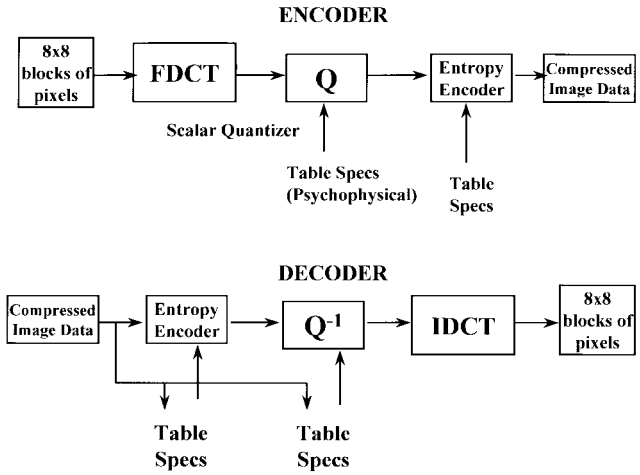

Fig. 19. Block diagram of JPEG encoder and decoder.

performs run-length coding on the resulting DCT sequences of coefficients (based on a Huffman coder), with the DC coefficients being represented in terms of their difference between adjacent blocks. Last, the compressed image data is transmitted over the channel to the image receiver. The decoder performs the inverse operations of the encoder.

a) Progressive encoding: Progressive encoding modes are provided within the JPEG syntax in order to provide a layering or progressive encoding capability to be applied to an image, i.e., to provide for an image to be transmitted at a low rate (and a low quality) and then progressively improved by subsequent transmissions. This capability is convenient for browsing applications where a low-quality or low-resolution image is more than adequate for things like scanning through the pages of a catalog.

Progressive encoding depends on being able to store the quantized DCT coefficients for an entire image. There are two forms of progressive encoding for JPEG, namely, spectral selection and successive approximation. For spectral selection, the initial transmission sends low-frequency DCT coefficients, followed progressively by the higher frequency coefficients, according to the zigzag scan used to order the DCT coefficients. Thus, the first transmission might send the lowest three DCT coefficients for all the 8×8 blocks of the image, followed by the next higher three DCT coefficients and so forth until all the DCT coefficients have been transmitted. The resulting scheme is simple to implement, but each image lacks the high-frequency components until the end layers are transmitted; hence, the reconstructed images from the early scans are blurred.

With the successive approximation method,[5] all the DCT coefficients for each 8×8 block are sent in each scan. Instead of sending them at full resolution, however, only the most significant bits of each coefficient are sent in the first scan, followed by the next most significant bits, and so on until all the bits are sent. The resulting reconstructed images are of reasonably good quality, even for the very early scans, since the high-frequency components of the image are preserved in all scans.

[5] Sometimes called SNR scalability.

The JPEG algorithm also supports a hierarchical or pyramid mode in which the image can be sent in one of several resolution modes to accommodate different types of displays. The way this is achieved is by filtering and downsampling the image in multiples of two in each dimension. The resulting decoded image is upsampled and subtracted from the next level, which is then coded and transmitted as the next layer. This process is repeated until all layers have been coded and transmitted.

The lossless mode of JPEG coding is different than the lossy mode shown in Fig. 19. Fundamentally, the image pixels are handled separately (i.e., the 8×8 block structure is not used), and each pixel is predicted based on three adjacent pixels using one of eight possible predictor modes. An entropy encoder is then used to encode the predicted pixels losslessly.

b) JPEG performance: If we assume that the pixels of an arbitrary color image are digitized to 8 b for luminance and 16 b for chrominance (where the chrominance signals are sampled at one-half the rate[6] of the luminance signal), then effectively there are 16 b/pixel that are used to represent an arbitrary color image. Using JPEG compression on a wide variety of such color images, the following image qualities have been measured subjectively:

Bits/pixel	Quality	Compression Ratio
≥ 2	Indistinguishable	8-to-1
1.5	Excellent	10.7-to-1
0.75	Very Good	21.4-to-1
0.50	Good	32-to-1
0.25	Fair	64-to-1.

The bottom line is that for many images, good quality can be obtained with about 0.5 b/pixel with JPEG providing a 32-to-1 compression.

4) JPEG-2000 Color Still-Image Coding [30]: Quite a lot of research has been undertaken on still-image coding since the JPEG standards were established in the early 1990's. JPEG-2000 is an attempt to focus these research efforts into a new standard for coding still color images.

The scope of JPEG-2000 includes not only potential new compression algorithms but also flexible compression architectures and formats. It is anticipated that an architecturally based standard has the potential of allowing the JPEG-2000 standard to integrate new algorithm components through downloaded software without requiring yet another new standards definition.

Some examples of the application areas for JPEG-2000 include:

- document imaging
- facsimile;
- Internet/WWW imagery;
- remote sensing;
- video component frames;
- photo and art digital libraries;

[6] The so-called 4:2:2 color sampling.

- medical imagery;
- security cameras;
- client-server;
- scanner/digital copiers;
- prepress;
- electronic photography.

JPEG-2000 is intended to provide low-bit-rate operation with subjective image-quality performance superior to existing standards without sacrificing performance at higher bit rates. It should be completed by the year 2000 and offer state-of-the-art compression for many years beyond.

JPEG-2000 will serve still-image compression needs that currently are not served. It will also provide access to markets that currently do not consider compression as useful for their applications. Specifically, it will address areas where current standards fail to produce the best quality or performance, including the following.

- *Low-bit-rate compression performance:* Current JPEG offers excellent compression performance in the middle and high bit rates above about 0.5 b/pixel. At low bit rates (e.g., below 0.25 b/pixel for highly detailed images), however, the distortion, especially when judged subjectively, becomes unacceptable compared with more modern algorithms such as wavelet subband coding [31].
- *Large images:* Currently, the JPEG image-compression algorithm does not allow for images greater then 64 $\times$ 64 K without tiling (i.e., processing the image in sections).
- *Continuous-tone and bilevel compression:* JPEG-2000 should be capable of compressing images containing both continuous-tone and bilevel images. It should also compress and decompress images with various dynamic ranges (e.g., 1–16 b) for each color component. Applications using these features include compound documents with images and text, medical images with annotation overlays, graphic and computer-generated images with binary and near-to-binary regions, alpha and transparency planes, and, of course, bilevel facsimile.
- *Lossless and lossy compression:* It is desired to provide lossless compression naturally in the course of progressive decoding (difference image encoding, or any other technique, which allows for the lossless reconstruction is valid). Applications that can use this feature include medical images, image archival applications where the highest quality is vital for preservation but not necessary for display, network applications that supply devices with different capabilities and resources, and prepress imagery.
- *Progressive transmission by pixel accuracy and resolution:* Progressive transmission that allows images to be reconstructed with increasing pixel accuracy or spatial resolution as more bits are received is essential for many applications. This feature allows the reconstruction of images with different resolutions and pixel accuracy, as needed or desired, for different target devices. Examples of applications include the World Wide Web, image archival applications, printers, etc.
- *Robustness to bit errors:* JPEG-2000 must be robust to bit errors. One application where this is important is wireless communication channels. Some portions of the bit stream may be more important than others in determining decoded image quality. Proper design of the bit stream can aid subsequent error-correction systems in alleviating catastrophic decoding failures. Usage of error confinement, error concealment, restart capabilities, or source-channel coding schemes can help minimize the effects of bit errors.
- *Open architecture:* It is desirable to allow open architecture to optimize the system for different image types and applications. This may be done either by the development of a highly flexible coding tool or adoption of a syntactic description language, which should allow the dissemination and integration of new compression tools. Work being done in MPEG-4 (see Section III-E2) on the development of downloadable software capability may be of use. With this capability, the user can select tools appropriate to the application and provide for future growth. With this feature, the decoder is only required to implement the core tool set plus a parser that understands and executes downloadable software in the bit stream. If necessary, unknown tools are requested by the decoder and sent from the source.
- *Sequential one-pass decoding capability (real-time coding):* JPEG-2000 should be capable of compressing and decompressing images with a single sequential pass. It should also be capable of processing an image using either component interleaved order or noninterleaved order. However, there is no requirement of optimal compression performance during sequential one-pass operation.
- *Content-based description:* Finding an image in a large data base of images is an important problem in image processing. This could have major applications in medicine, law enforcement, environment, and image archival applications. A content-based description of images might be available as a part of the compression system. JPEG-2000 should strive to provide the opportunity for solutions to this problem.
- *Image security:* Protection of the property rights of a digital image can be achieved by means of watermarking, labeling, stamping, encryption, etc. Watermarking is an invisible mark inside the image content. Labeling is already implemented in still-picture interchange file format and must be easy to transfer back and forth to a JPEG-2000 image file. Stamping is a very visible and annoying mark overlayed onto a displayed image that can only be removed by a specific process. Encryption can be applied on the whole image file or limited to

part of it (header, directory, image data) in order to avoid unauthorized use of the image.

- *Side-channel spatial information (transparency):* Side-channel spatial information, such as alpha planes and transparency planes, are useful for transmitting information for processing the image for display, print, editing, etc. An example of this is the transparency plane used in World Wide Web applications.

For JPEG-2000, a prime candidate for the base signal processing would seem to be wavelet subband coding. Compared with the DCT as used in JPEG coding, wavelet coding is able to achieve the advantages of low-bit-rate coding with large block size while at the same time providing progressive transmission and scalability features. However, the low-pass wavelet filter may not be optimum in terms of picture quality versus bandwidth. Thus, another candidate might be MPEG intracoding with the pyramid-style progressive transmission found in JPEG. With pyramid coding, the filtering can be optimized since it is independent of the coding.

E. Video Coding [32], [33]

Video signals differ from image signals in several important characteristics. Of course, the most important difference is that video signals have an associated frame rate of anywhere from 15 to 60 frames/s, which provides the illusion of motion in the displayed signal. A second difference is the frame size. Video signals may be as small as QCIF (176 $\times$ 144 pixels) and as large as HDTV (1920 $\times$ 1080 pixels), whereas still images are sized primarily to fit PC color monitors (640 $\times$ 480 pixels for VGA or 1024 $\times$ 768 pixels for XVGA). A third difference between images and video is the ability to exploit temporal masking as well as spectral masking in designing compression methods for video. Hence, a moving object in a scene tends to mask the background that emerges when the object moves, making it easier to compress that part of the uncovered image. Last, one can also take advantage of the fact that objects in video sequences tend to move in predictable patterns and can therefore be "motion compensated" from frame to frame if we can reliably detect both the object and its motion trajectory over time.

Fundamentally, there have been five major initiatives in video coding that have led to a range of video standards:

- video coding for video teleconferencing, which has led to ITU standards H.261 for ISDN video conferencing [34], H.263 for POTS video conferencing [35], and H.262 for ATM/broad-band video conferencing;
- video coding for storing movies on CD read-only memory (ROM), with on the order of 1.2 Mb/s allocated to video coding and 256 kb/s allocated to audio coding, which led to the initial ISO-MPEG-1 standard;
- video coding for storing broadcast video on DVD, with on the order of 2–15 Mb/s allocated to video and audio coding, which led to the ISO-MPEG-2 standard;
- video coding for low-bit-rate video telephony over POTS networks, with as little as 10 kb/s allocated to video and as little as 5.3 kb/s allocated to voice coding, which led to the H.324 standard and will ultimately lead to the ISO-MPEG-4 standard [36];
- video coding for advanced HDTV, with 15–400 Mb/s allocated to the video coding.

In the following sections, we provide brief summaries of each of these video coders, with the goal of describing the basic coding algorithm as well as the features that support the use of video coding in multimedia applications.

1) H.261 and Its Derivatives: The H.261 video codec, initially intended for ISDN teleconferencing, is the baseline video mode for most multimedia conferencing systems. The H.262 video codec is essentially the high-bit-rate MPEG-2 standard and will be described later in this paper. The H.263 low bit rate video codec is intended for use in POTS teleconferencing at modem rates of 14.4–56 kb/s, where the modem rate includes video coding, speech coding, control information, and other logical channels for data.

The H.261 codec codes video frames using a DCT on blocks of size 8 $\times$ 8 pixels, much the same as used for the JPEG coder described previously. An initial frame (called an INTRA frame) is coded and transmitted as an independent frame. Subsequent frames, which are modeled as changing slowly due to small motions of objects in the scene, are coded efficiently in the INTER mode using a technique called motion compensation, in which the displacement of groups of pixels from their position in the previous frame (as represented by so-called motion vectors) is transmitted together with the DCT coded difference between the predicted and original images [37].

Since H.261 is intended for conferencing applications with only small, controlled amounts of motion in a scene, and with rather limited views consisting mainly of head-and-shoulders views of people along with the background, the video formats that are supported include both the CIF and the QCIF format. All H.261 video is noninterlaced, using a simple progressive scanning pattern.

A unique feature of H.261 is that it specifies a standard coded video syntax and decoding procedure, but most choices in the encoding methods, such as allocation of bits to different parts of the picture, are left open and can be changed by the encoder at will. The result is that the quality of H.261 video, even at a given bit rate, depends greatly on the encoder implementation. This explains why some H.261 systems appear to work better than others.

Since motion compensation is a key element in most video coders, it is worthwhile understanding the basic concepts in this processing step. Fig. 20 shows a block diagram of a motion-compensated image coder. The key idea is to combine transform coding (in the form of the DCT of 8 $\times$ 8 pixel blocks) with predictive coding (in the form of differential PCM) in order to reduce storage and computation of the compressed image and at the same time to give a high degree of compression and adaptability. Since motion compensation is difficult to perform in the transform domain, the first step in the interframe coder is to create a motion-compensated prediction error using

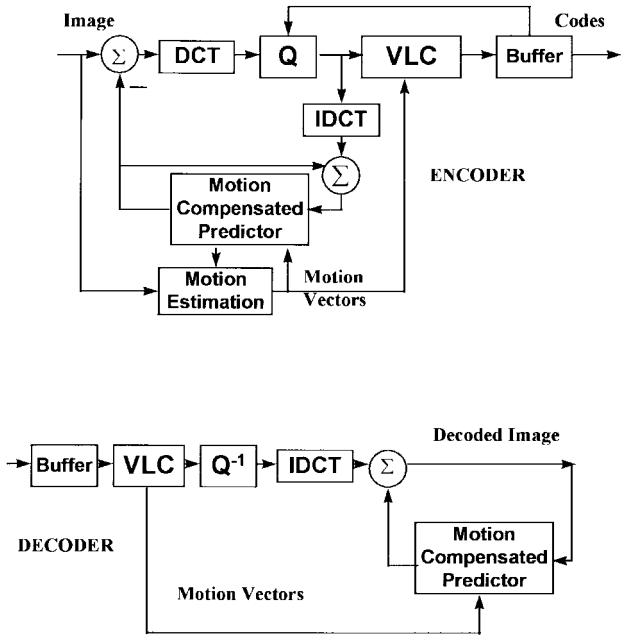

Fig. 20. Motion-compensated coder for interframe coding.

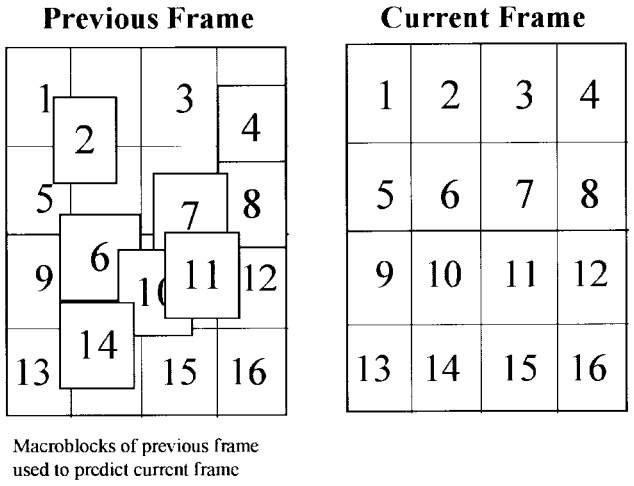

Fig. 21. Illustration of motion-compensated coding of interframe macroblocks.

macroblocks of 16 × 16 pixels. This computation requires only a single frame store in the receiver. The resulting error signal is transformed using a DCT, quantized by an adaptive quantizer, entropy encoded using a variable length coder, and buffered for transmission over a fixed-rate channel.

The way that the motion estimator works is illustrated in Fig. 21. Each 16 × 16-pixel macroblock in the current frame is compared with a set of macroblocks in the previous frame to determine the one that best predicts the current macroblock. The set of macroblocks includes those within a limited region of the current macroblock. When the best matching macroblock is found, a motion vector is determined, which specifies the reference macroblock and the direction of the motion of that macroblock.

a) H.263 coding: The H.263 video codec is based on the same DCT and motion-compensation techniques used in H.261. Several small, incremental improvements in video coding were added to the H.263 standard for use in POTS conferencing. These included the following:

- half-pixel motion compensation in order to reduce the roughness in measuring best matching blocks with coarse time quantization; this feature significantly improves the prediction capability of the motion-compensation algorithm in cases where there is object motion that needs fine spatial resolution for accurate modeling;
- improved variable-length coding;
- reduced overhead of channel input;
- optional modes, including unrestricted motion vectors that are allowed to point outside the picture;
- arithmetic coding in place of the variable-length (Huffman) coding;
- advanced motion prediction mode, including overlapped block motion compensation;
- a mode that combines a bidirectionally predicted picture with a normal forward predicted picture.

In addition, H.263 supports a wider range of picture formats, including 4CIF (704 × 576 pixels) and 16CIF (1408 × 1152 pixels), to provide a high-resolution-mode picture capability.

2) MPEG-1 Video Coding: The MPEG-1 standard is a true multimedia standard with specifications for coding, compression, and transmission of audio, video, and data streams in a series of synchronized, mixed packets. The driving focus of the standard was storage of multimedia content on a standard CD-ROM, which supported data-transfer rates of 1.4 Mb/s and a total storage capability of about 600 MB. The picture format that was chosen was the source input format (352 × 288 pixels at 25 noninterlaced frames/s or 352 × 240 pixels at 30 noninterlaced frames/s), which was intended to provide VHS VCR-like video and audio quality along with VCR-like controls.

The video coding in MPEG-1 is very similar to the video coding of the H.26X series described above. Both systems process the video signal by doing spatial coding using a DCT of 8 × 8 pixel blocks, quantizing the DCT coefficients based on perceptual weighting criteria, storing the DCT coefficients for each block in a zigzag scan, and doing a variable run-length coding of the resulting DCT coefficient stream. Temporal coding is achieved by using the ideas of uni- and bidirectional motion-compensated prediction, with three types of pictures resulting, namely:

- "I" or intrapictures, which are coded independently of all previous or future pictures;
- "P" or predictive pictures, which are coded based on previous I or previous P pictures;
- "B" or bidirectionally predictive pictures, which are coded based on the next and/or the previous pictures.

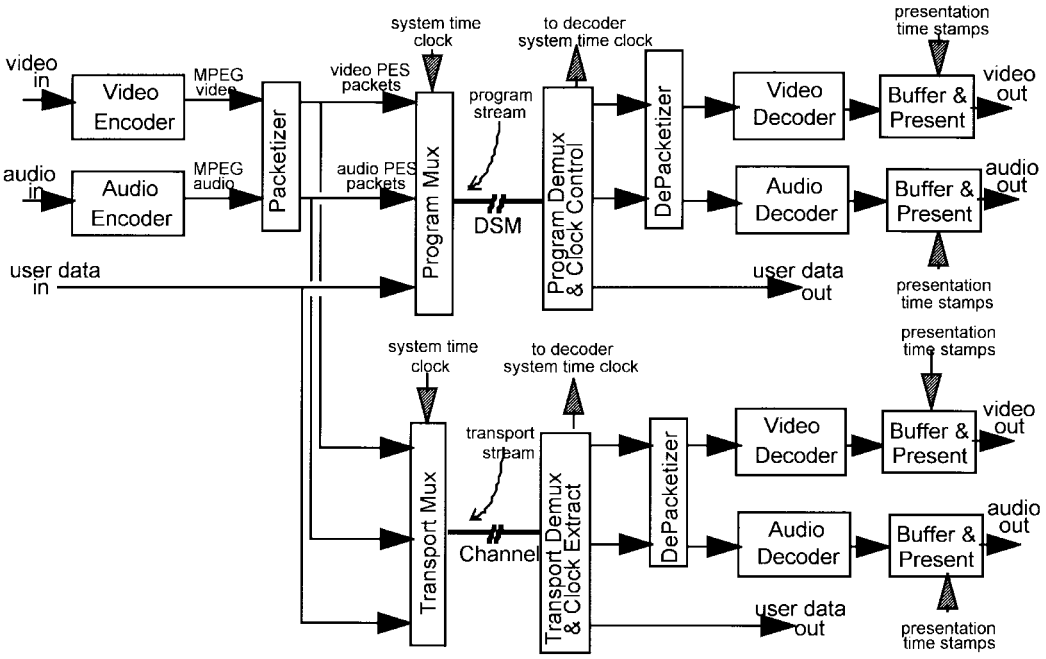

Fig. 22. MPEG-2 systems multiplex showing program (upper half of figure) and transport (lower half of figure) streams.

High-quality audio coding is an implicit part of the MPEG-1 standard, and therefore it includes sampling rates of 32, 44.1, and 48 kHz, thereby providing provision for near-CD audio quality.

3) MPEG-2 Coding: The MPEG-2 standard was designed to provide the capability for compressing, coding, and transmitting high-quality, multichannel, multimedia signals over broad-band networks, for example, using ATM protocols. The MPEG-2 standard specifies the requirements for video coding, audio coding, systems coding for combining coded audio and video with user-defined private data streams, conformance testing to verify that bit streams and decoders meet the requirements, and software simulation for encoding and decoding of both the program and the transport streams. Because MPEG-2 was designed as a transmission standard, it supports a variety of packet formats (including long and variable-length packets of from 1 kB up to 64 kB) and provides error-correction capability that is suitable for transmission over CATV and satellite links.

a) MPEG-2 systems: The MPEG-2 systems level defines two types of streams: the program stream and the transport stream. The program stream is similar to that used in MPEG-1 but with a modified syntax and new functions to support advanced functionalities. Program-stream decoders typically use long and variable-length packets. These are well suited for software-based processing and error-free environments. The transport streams offer robustness necessary for noisy channels and also provide the ability to include multiple programs in a single stream. The transport stream uses fixed-length packets (of size 188 bytes). It is well suited for delivering compressed video and audio over error-prone channels such as CATV networks and satellite transponders. A block diagram of the MPEG-2 systems multiplex showing program and transport streams is given in Fig. 22.

The basic data structure that is used for both the program stream and the transport stream data is called the packetized elementary stream (PES) packet. PES packets are generated using the compressed video and audio data. A program stream is generated by interleaving PES packets from the various encoders with other packets containing necessary data to generate a single bit stream. A transport stream contains packets of fixed length consisting of 4 bytes of header followed by 184 bytes of data obtained by chopping up the data in the PES packets. The key difference in the streams is that the program streams are intended for error-free environments, whereas the transport streams are intended for noisier environments where some type of error protection is required.

b) MPEG-2 video: MPEG-2 video was originally designed for high-quality encoding of interlaced video from standard TV with bit rates on the order of 4–9 Mb/s. Over time, the MPEG-2 video standard was expanded to include high-resolution video, such as HDTV, as well as hierarchical or scalable video coding. Since MPEG-2 video does not standardize the encoding method but only the video bit-stream syntax and decoding semantics, there have evolved two generalized video codecs, one for nonscalable video coding and one for scalable video coding. Fig. 23 shows a block diagram of the MPEG-2 nonscalable video-coding algorithm. The video encoder consists of an interframe/field DCT encoder, a frame/field motion estimator and compensator, and a variable-length encoder. The frame/field DCT encoder exploits spatial redundancies in the video, and the frame/field motion compensator exploits temporal redundancies in the video signal. The coded video bit stream is sent to a systems

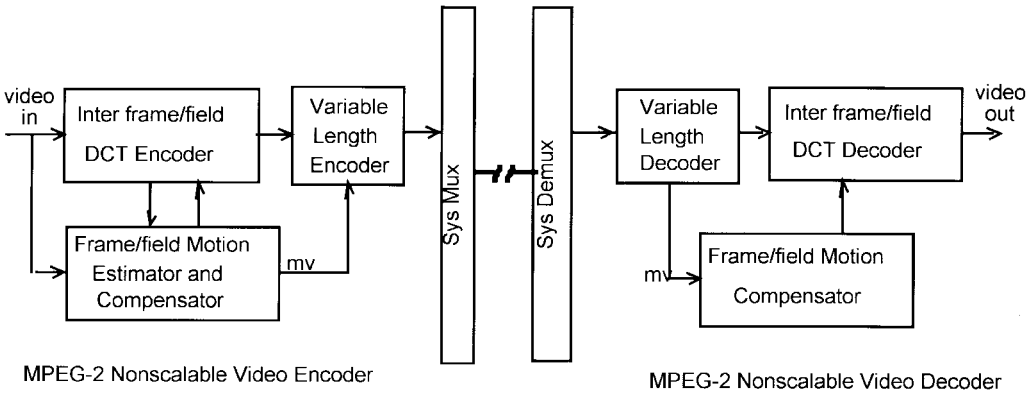

Fig. 23. A generalized codec for MPEG-2 nonscalable video coding.

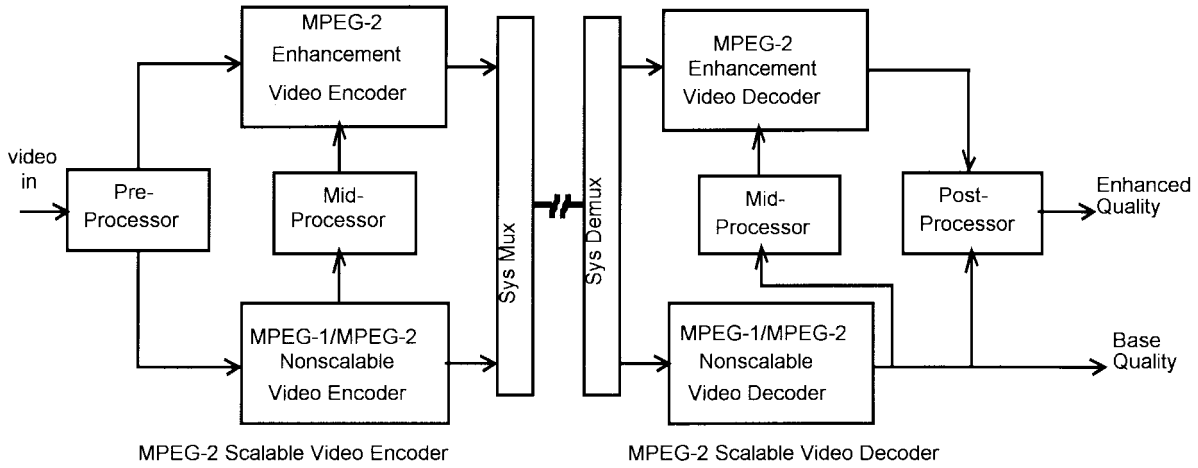

Fig. 24. A generalized codec for MPEG-2 scalable video coding.

multiplexer, Sys Mux, which outputs either a transport or a program stream.

The MPEG-2 decoder of Fig. 23 consists of a variable-length decoder (VLD), interframe/field DCT decoder, and frame/field motion compensator. Sys Demux performs the complementary function of Sys Mux and presents the video bit stream to VLD for decoding of motion vectors and DCT coefficients. The frame/field motion compensator uses the motion vectors decoded by the VLD to generate a motion-compensated prediction that is added back to the decoded prediction error signal to generate the decoded video out. This type of coding produces nonscalable video bit streams since, normally, the full spatial and temporal resolution coded is the one that is expected to be decoded.

A block diagram of a generalized codec for MPEG-2 scalable video coding is shown in Fig. 24. Scalability is the property that allows decoders of various complexities to be able to decode video of resolution/quality commensurate with their complexity from the same bit stream. The generalized structure of Fig. 24 provides capability for both spatial and temporal resolution scalability in the following manner. The input video goes through a preprocessor that produces two video signals, one of which (called the base layer) is input to a standard MPEG-1 or MPEG-2 nonscalable video encoder and the other (called the enhancement layer) of which is input to an MPEG-2 enhancement video encoder. The two bit streams, one from each encoder, are multiplexed in Sys Mux (along with coded audio and user data). In this manner, it becomes possible for two types of decoders to be able to decode a video signal of quality commensurate with their complexity from the same encoded bit stream.

4) MPEG-4 Methods: Most recently, the focus of video coding has shifted to *object-based* coding at rates of 8 kb/s or lower and 1 Mb/s or higher. Key aspects of this newly proposed MPEG standard include independent coding of objects in a picture; the ability to interactively composite these objects into a scene at the display; the ability to combine graphics, animated objects, and natural objects in the scene; and the ability to transmit scenes in higher dimensionality formats (e.g., 3-D). Also inherent in the MPEG-4 standard is the concept of video scalability, both in the temporal and spatial domains, in order to effectively control the video bit rate at the transmitter, in the network, and at the receiver so as to match the available transmission and processing resources. MPEG-4 is scheduled to be complete in 1998.

MPEG-4 builds on and combines elements from three fields: digital television, interactive graphics, and the World Wide Web. It aims to provide a merging of the production,

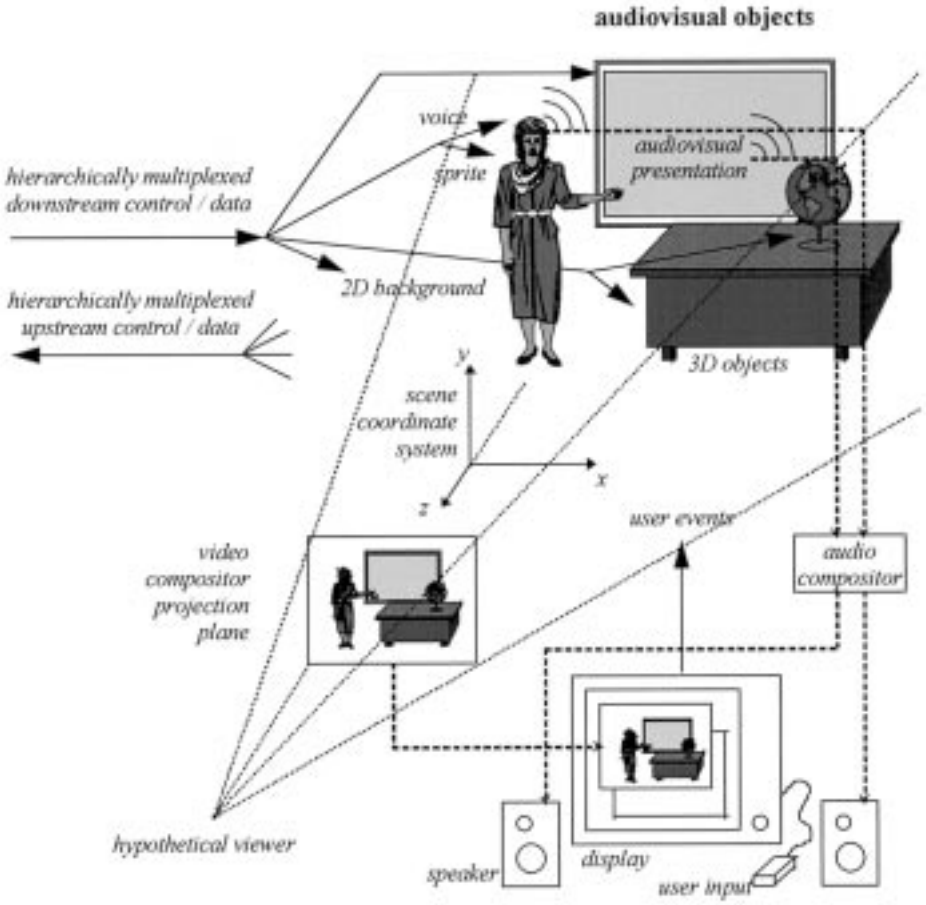

Fig. 25. An example of an MPEG-4 scene. (Courtesy of MPEG Systems Group.)

distribution, and display elements of these three fields. In particular, it is expected that MPEG-4 will provide:

- multimedia content in a form that is reusable, with the capability and flexibility of incorporating on-the-fly piece parts from anywhere and at any time the application desires;
- protection mechanisms for intellectual property rights associated with that content;
- content transportation with a QoS custom tailored to each component;
- high levels of user interaction, with some control features being provided by the multimedia signal itself and others available locally at the receiving terminal.

The design of MPEG-4 is centered around a basic unit of content called the *audio-visual object* (AVO). Examples of AVO's are a musician (in motion) in an orchestra, the sound generated by that musician, the chair she is sitting on, the (possibly moving) background behind the orchestra, explanatory text for the current passage, etc. In MPEG-4, each AVO is represented separately and becomes the basis for an independent stream.

For a viewer to receive a selection that can be seen on a display and heard through loudspeakers, the AVO's must be transmitted from a storage (or live) site. Since some AVO's may have an extremely long duration, it is usually undesirable to send each one separately, in its entirety, one after the other. Instead, some AVO's are *multiplexed* together and sent simultaneously so that replay can commence shortly after transmission begins. Other AVO's needing a different QoS can be multiplexed and sent on another transmission path that is able to provide that QoS.

Upon arrival of the AVO's, they must be assembled or *composed* into an audio-visual *scene*. In its most general form, the scene may be 3-D. Since some of the AVO's, such as moving persons or music, involve real-time portrayal, proper time synchronization must also be provided.

AVO's can be arbitrarily composed. For example, the musicians could be moved around to achieve special effects, e.g., one could choose to see and hear only the trumpet section. Alternatively, it would be possible to delete the drummer (and the resulting drum audio component), leaving the rest of the band so that the viewer could play along on his own drums. Fig. 25 illustrates scene composition in MPEG-4.

Following composition of a 3-D scene, the visual AVO's must be projected onto a viewer plane for display, and the audio AVO's must be combined for playing through loudspeakers or headphones. This process is called *rendering.*

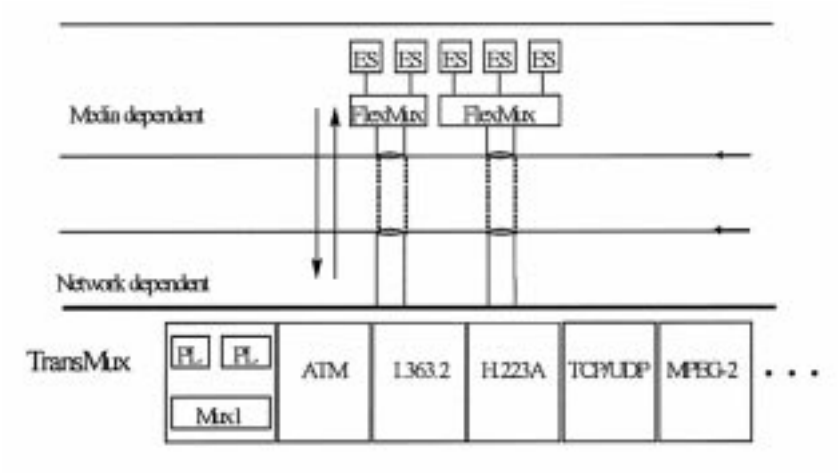

Fig. 26. The MPEG-4 two-layer multiplex. (Courtesy of MPEG Systems Group.)

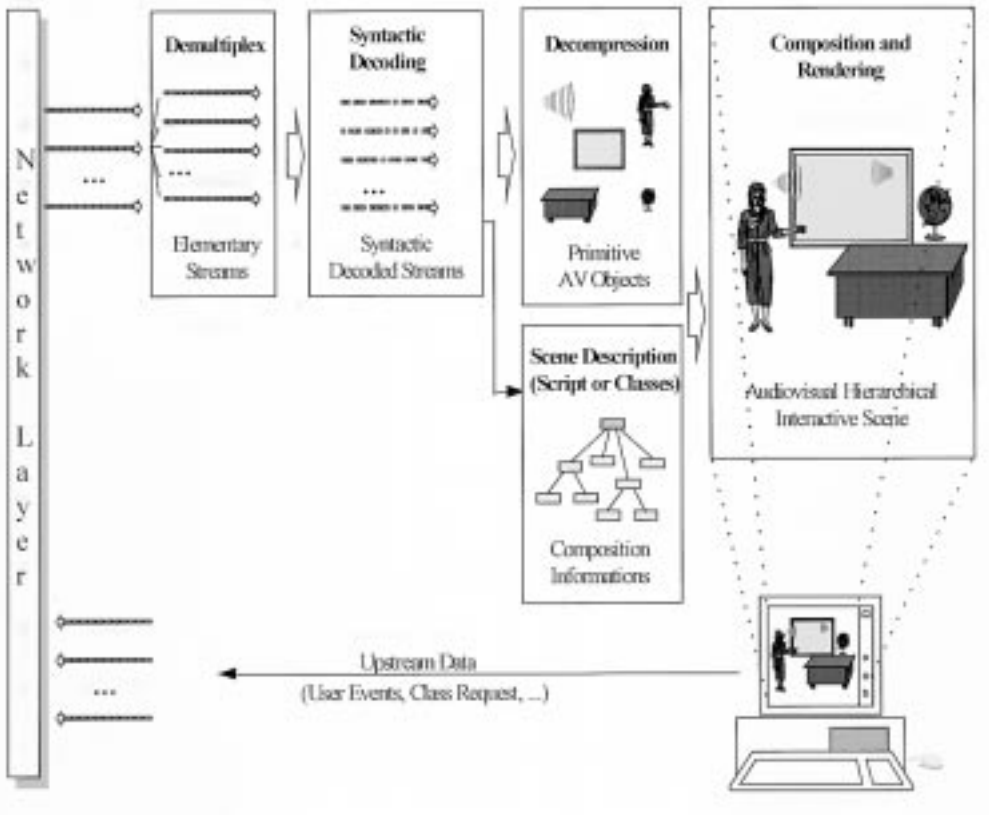

Fig. 27. Major components of an MPEG-4 terminal (receiver side). (Courtesy of MPEG Systems Group.)

In principle, rendering does not require standardization. All that is needed is a viewpoint and a window size.

a) MPEG-4 multiplex: The transmission of coded real-time AVO's from one or more sources to a destination is accomplished through the two-layer multiplex shown in Fig. 26. Each coded elementary AVO is assigned to an elementary stream (ES). The FlexMux layer then groups together ES's having similar QoS requirements to form FlexMux Streams. The TransMux layer then provides transport services matched to the required QoS of each FlexMux stream. TransMux can be any of the existing transport protocols such as universal datagram packet (UDP)/IP, advanced adaptation layer 5/ATM, MPEG-2 transport stream, etc. The multiplexing standard has not yet been approved by the MPEG-4 committee.

b) MPEG-4 systems: The systems part of MPEG-4 specifies the overall architecture of a general receiving terminal. Fig. 27 shows the major elements. FlexMux streams coming from the network are passed to appropriate FlexMux demultiplexers that produce ES's. The ES's are then syntactically decoded into intermediate data, such as motion vectors and DCT coefficients, and then passed to

appropriate decompressors that produce the final AVO's, which are composed and rendered into the final display.

To place the AVO's into a scene (composition), their spatial and temporal relationships (the scene structure) must be known. For example, the scene structure may be defined by a multimedia author or interactively by the end viewer. Alternatively, it could be defined by one or more network elements that manage multiple sources and the multipoint communication between them. In any event, the composition part of MPEG-4 systems specifies the methodology for defining this structure.

Temporally, all AVO's have a single time dimension. For real-time, high-quality operation, end-to-end delay from the encoder input to the decoder output should be constant. At low bit rates or operation over lossy networks, however, the ideal of constant delay may have to be sacrificed. This delay is the sum of the delays from encoding (including video frame dropping), encoder buffering, multiplexing, communication or storage, demultiplexing, decoder buffering, decoding (including frame repeating), and presentation.

The transmitted data streams must contain either implicit or explicit timing information. As in MPEG-1 and MPEG-2, there are two kinds of timing information. One indicates periodic values of the encoder clock, while the other tells the desired presentation timing for each AVO. Either one is optional and, if missing, must be provided by the receiver compositor.

Spatially, each AVO has its own *local coordinate system*, which serves to describe local behavior independent of the scene or any other AVO's. AVO's are placed in a scene by specifying (possibly dynamic) coordinate transformations from the local coordinate systems into a common scene coordinate system, as shown in Fig. 25. Note that the coordinate transformations, which position AVO's in a scene, are part of the scene structure, not the AVO. Thus, object motion in the scene is the motion specified locally by the AVO plus the motion specified by the dynamic coordinate transformations.

The scene description is sent as a separate elementary stream. This allows for relatively simple bit-stream editing, one of the central functionalities in MPEG-4. In bit-stream editing, we want to be able to change the composition and scene structure without decoding the AVO bit streams and changing their content.

To increase the power of editing and scene manipulation even further, the MPEG-4 scene structure may be defined hierarchically and represented as a tree. Each node of the tree is an AVO, as illustrated in Fig. 28. Nodes at the leaves of the tree are *primitive nodes.* Nodes that are parents of one or more other nodes are *compound nodes.* Primitive nodes may have elementary streams assigned to them, whereas compound nodes are of use mainly in editing and compositing.

In the tree, each AVO is positioned in the local coordinate system of its parent AVO. The tree structure may be dynamic, i.e., the positions can change with time, and nodes may be added or deleted. The information describing the relationships between parent nodes and children nodes

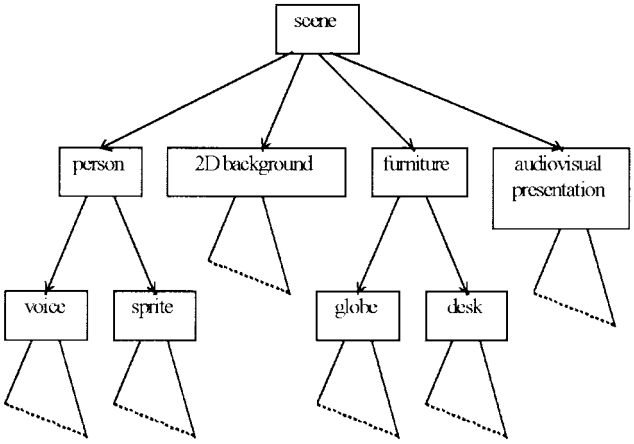

Fig. 28. Logical structure of the scene. (Courtesy of MPEG Systems Group.)

is sent in the elementary stream assigned to the scene description.

c) MPEG-4 audio: A number of functionalities are provided to facilitate a wide variety of applications that could range from low-quality but intelligible speech to high-quality, multichannel audio. Examples of the functionalities include speed control, pitch change, error resilience and scalability in terms of bit rate, bandwidth, error robustness, complexity, etc., as defined below.

- The speed-change functionality allows the change of the time scale without altering the pitch during the decoding process. This can, for example, be used to implement a "fast-forward" function (audio scan) or to adapt the length of an audio sequence to a given video sequence.
- The pitch-change functionality allows the change of the pitch without altering the time scale during the encoding or decoding process. This can be used, for example, for voice alteration or karaoke-type applications.
- Bit-rate scalability allows a bit stream to be parsed into a bit stream of lower bit rate such that the combination can still be decoded into a meaningful signal. The bit stream parsing can occur either during transmission or in the decoder.
- Bandwidth scalability is a particular case of bit-rate scalability, whereby part of a bit stream representing a part of the frequency spectrum can be discarded during transmission or decoding.
- Encoder complexity scalability allows encoders of different complexity to generate valid and meaningful bit streams.
- Decoder complexity scalability allows a given bit stream to be decoded by decoders of different levels of complexity. The audio quality, in general, is related to the complexity of the encoder and decoder used.
- Error robustness provides the ability for a decoder to avoid or conceal audible distortion caused by transmission errors.

- Hybrid scalable text-to-speech (TTS) synthesis provides the ability to generate synthetic speech from textual AVO's, including both audio and visual (talking head) components.
- Basic music instrument digital interface support enables musical instruments to be added to the composite scene at the decoder using low-data-rate descriptions of the music and the instruments.
- Basic synthetic audio description allows low-level audio signals (e.g., beeps, sound effects, background sounds) to be added at the decoder.

d) Natural 2-D motion video: MPEG-4 coding for natural video will, of course, perform efficient compression of traditional video-camera signals for storage and transmission in multimedia environments. However, it will also provide tools that enable a number of other functionalities such as object scalability, spatial and temporal scalability, sprite (or avatar) overlays, error resilience, etc. MPEG-4 video will be capable of coding conventional rectangular video as well as arbitrarily shaped 2-D objects in a video scene. The MPEG-4 video standard will be able to code video ranging from very low spatial and temporal resolutions in progressive scanning format up to very high spatial and temporal resolutions for professional studio applications, including interlaced video. The input frame rate can be nonuniform, including single-picture input, which is considered as a special case.

The basic video AVO is called a video object (VO). If the VO is *scalable,* it may be split up, coded, and sent in two or more VO layers (VOL's). One of these VOL's is called the base layer, which all terminals must receive in order to display any kind of video. The remaining VOL's are called enhancement layers, which may be expendable in case of transmission errors or restricted transmission capacity. For example, in a broadcast application, transmitting to a variety of terminals having different processing capabilities or whose connections to the network are at different bit rates, some of the receiving terminals might receive all of the VOL's, while others may receive only a few, while still others may receive only the base-layer VOL.

In a scalable video object, the VO is a compound AVO that is the parent of two or more child VOL's. Each VOL is a primitive AVO and is carried by a separate elementary stream.

A snapshot in time of a VOL is called a video object plane (VOP). For rectangular video, this corresponds to a picture in MPEG-1 and MPEG-2 or a frame in other standards. In general, however, the VOP can have arbitrary shape.

The VOP is the basic unit of coding and is made up of luminance (Y) and chrominance (Cb, Cr) components plus shape information. The shape and location of VOP's may vary from one VOP to the next. The shape may be conveyed either implicitly or explicitly. With implicit shape coding, the irregularly shaped object is simply placed in front of a colored (say, blue-green) background known to the receiver, and a rectangular VOP containing both object and background is coded and transmitted. The decoder retrieves the object by simple chroma-keying, as done in today's broadcast TV studios.

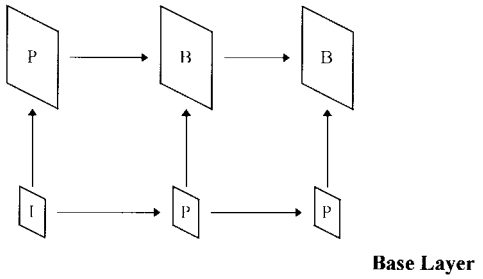

Fig. 29. Spatial scalability with two layers. (Courtesy of MPEG Video Group.)

Explicit shape is represented by a rectangular alpha plane that covers the object to be coded. An alpha plane may be *binary* ("0" for background, "1" for object) if only the shape is of interest, or it may be *gray level* (up to 8 b per pixel) to indicate various levels of partial transparency for the object. If the alpha plane has a constant gray value inside the object area, that value can be sent separately and the alpha plane coded as a binary alpha plane. Various algorithms for coding alpha planes are currently under study, the most promising of which is simple arithmetic coding of the bit planes.

Coding of texture for arbitrarily shaped regions whose shape is described with an alpha map is different than traditional methods. Techniques are borrowed from both H.263 and earlier MPEG standards. For example, intraframe coding, forward prediction motion compensation, and bidirectional motion compensation are used. This gives rise to the definitions of I-VOP's, P-VOP's, and B-VOP's for VOP's that are intracoded, forward predicted or bidirectionally predicted, respectively.

e) Multifunctional coding tools and algorithms: Multifunctional coding refers to features other than coding efficiency. For example, object-based spatial and temporal scalabilities are provided to enable broad-based access over a variety of networks and facilities. This can be useful for Internet and data base applications. Also, for mobile multimedia applications, spatial and temporal scalabilities are extremely useful for channel bandwidth scaling for robust delivery. Spatial scalability with two layers is shown in Fig. 29. Temporal scalability with two layers is shown in Fig. 30.

Multifunctional coding also addresses multiview and stereoscopic applications as well as representations that enable simultaneous coding and tracking of objects for surveillance and other applications. Besides the aforementioned applications, a number of tools are being developed for segmentation of a video scene into objects and for coding noise suppression.

f) Error resilience: Error resilience is needed, to some extent, in all transmission media. In particular, due to the rapid growth of mobile communications, it is extremely important that audio and video information is sent successfully via wireless networks. These networks are typically

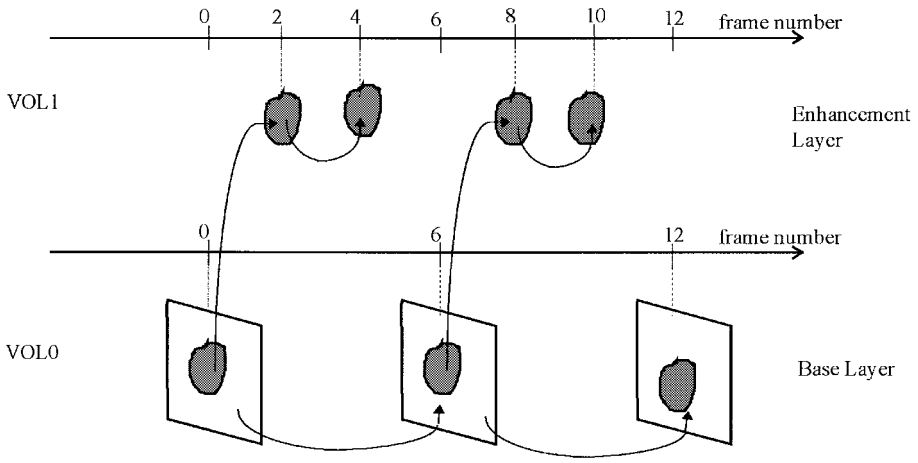

Fig. 30. Temporal scalability with two layers. (Courtesy of MPEG SNHC Group.)

prone to error and usually operate at relatively low bit rates, e.g., less than 64 kb/s. Both MPEG and the ITU-T are working on error resilience methods, including forward error correction, automatic request for retransmission, scalable coding, slice-based bit-stream partitioning, and motion-compensated error correction.

g) Synthetic images: Several efforts are under way to provide synthetic image capabilities in MPEG-4. There is no wish to reinvent existing graphics standards. Thus, MPEG-4 uses the virtual-reality modeling language as a starting point for its synthetic image specification. MPEG-4 will add a number of additional capabilities.

The first addition is a synthetic *face and body* (FAB) animation capability, which is a model-independent definition of artificial face and body animation parameters. With these parameters, one can represent facial expressions, body positions, mouth shapes, etc. Planned capabilities include 3-D feature point positions, 3-D head and body control meshes for animation, and texture mapping of face, body, and personal characteristics. Also planned is a text-driven mouth animation to be combined with a text-to-speech capability for a complete text-to-talking-head implementation. The head points are shown in Fig. 31.

Another capability being studied is for texture mapping of real image information onto artificial models such as the FAB model. For this, wavelet-based texture coding is being considered. An advantage of wavelet-based coding is the relative ease of adjusting the resolution of the visual information to match the requirements of the rendering. For example, if an object is being composed at a distance far from the viewer, then it is not necessary to send the object with high resolution.

Associated with FAB coding is a triangular mesh modeling capability to handle any type of 2- or 3-D synthetic or natural shape. This also facilitates integration of text and graphics onto synthetic and natural imagery. For example, putting text onto a moving natural object requires a tracking of features on the natural object, which is made easier by a mesh-based representation of the object.

h) Conclusion of MPEG-4 and launching of MPEG-7: In summary, MPEG-4 will integrate most of the capabilities and features of multimedia into one standard, including live audio and video, synthetic objects, and text, all of which can be combined on the fly. Multipoint conversations can be facilitated by displays tailored to each viewer or group of viewers. Multimedia presentations can be sent to auditoriums, offices, homes, and mobiles with delivery scaled to the capabilities of the various receivers.

However, MPEG-4 is not the end of the roadmap for multimedia standards. Plans are now under way to begin MPEG-7, which is aimed at defining multimedia features for purposes such as searching and browsing large data bases, identification, authentication, cataloging, etc. This phase of MPEG is scheduled for completion sometime around the early part of the twenty-first century.

5) HDTV—The Ultimate TV Experience: HDTV is designed to be the ultimate television-viewing experience, providing, in a cost-effective manner, a high-resolution and wide-screen television system with a more panoramic aspect ratio (16:9 versus the conventional 4:3 ratio for NTSC or PAL) and producing much higher quality pictures and sound. HDTV systems will be one of the first systems that will not be backward compatible with NTSC or PAL. The key enablers of HDTV systems are the advanced video and audio compression capability, the availability of inexpensive and powerful VLSI chips to realize the system, and the availability of large displays.

The primary driving force behind HDTV is picture and sound quality that can be received by the consumer in his home. This is achieved by increasing the spatial resolution by about a factor of two in both the horizontal and vertical dimensions, providing a picture with about 1000 scan lines and with more than 1000 pixels per scan line. In addition, HDTV will eventually use only progressive (noninterlaced) scanning at about 60 frames/s to allow better fast-action sports and far better interoperability with computers while eliminating the artifacts associated with interlaced pictures. Unfortunately, increasing the spatial and temporal resolution of the HDTV signal and adding multichannel sound also increases its analog bandwidth. Such an analog signal cannot be accommodated in a single channel of the currently allocated broadcast spectrum. Moreover,

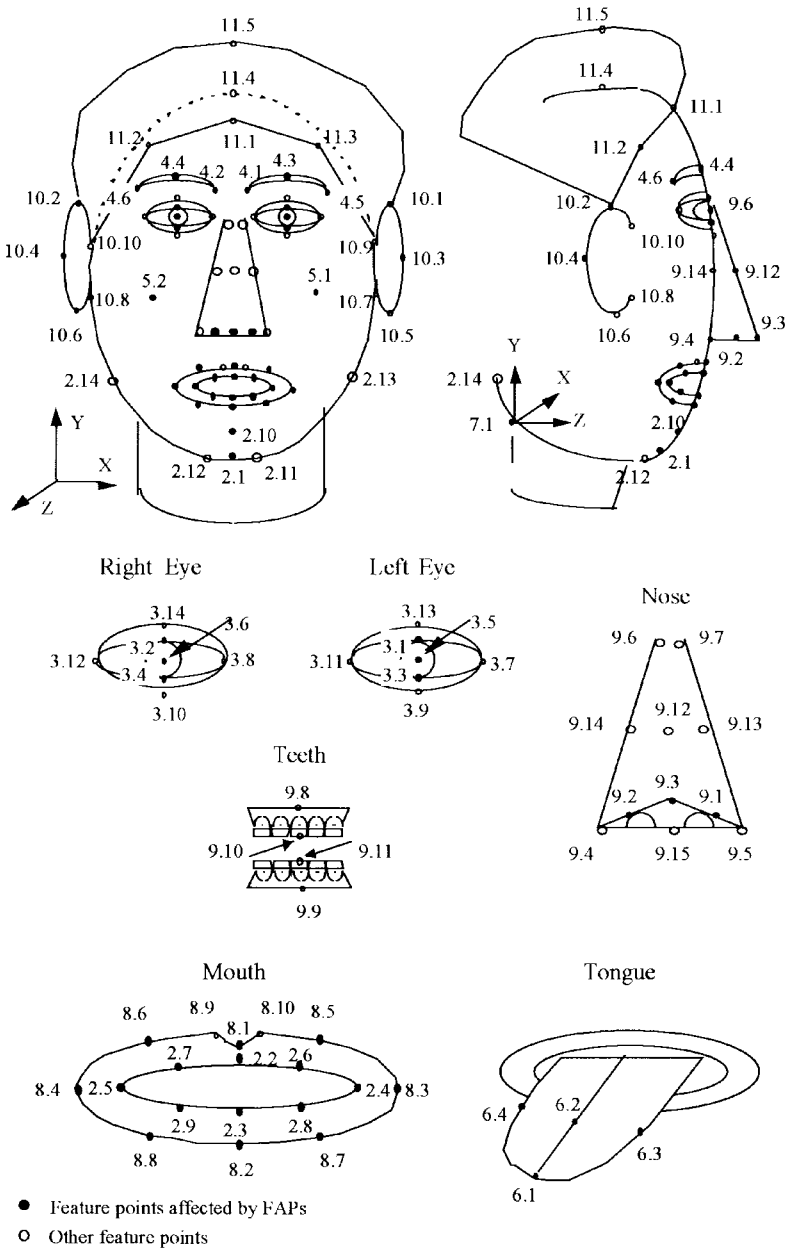

Fig. 31. Head control points for FAB animation. (Courtesy of MPEG SNHC Group.)

even if bandwidth were available, such an analog signal would suffer interference both to and from the existing TV transmissions. In fact, much of the available broadcast spectrum can be characterized as a fragile transmission channel. Many of the 6-MHz TV channels are kept unused because of interference considerations and are designated as *taboo* channels.

Therefore, all the current HDTV proposals employ digital compression, which reduces the bit rate from approximately 1 Gb/s to about 20 Mb/s, a rate that can be accommodated in a 6-MHz channel either in a terrestrial broadcast spectrum or a CATV channel. This digital signal is incompatible with the current television system and therefore can be decoded only by a special decoder.

The result of all these improvements is that the number of active pixels in an HDTV signal increases by about a factor of five (over NTSC signals), with a corresponding increase in the analog bandwidth or digital rate required to represent the uncompressed video signal. The quality of the audio associated with HDTV is also improved by means of multichannel, CD-quality surround sound, in which each channel is independently transmitted.

Since HDTV will be digital, different components of the information can be simply multiplexed in time instead of frequency multiplexed on different carriers, as in the case of analog TV. For example, each audio channel is independently compressed, and these compressed bits are multiplexed with compressed bits from video as well as bits for closed captioning, teletext, encryption, addressing, program identification, and other data services in a layered fashion.

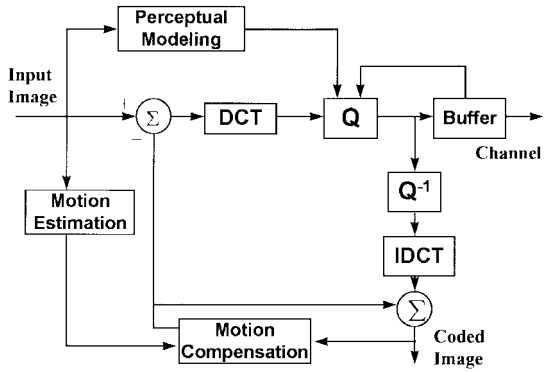

Fig. 32. Simplified HDTV video encoder.

a) HDTV video encoder: Fig. 32 shows a simplified block diagram of an HDTV video encoder that uses hierarchical, subpixel motion estimation, with perceptual coding of interframe differences, a fast response to scene and channel changes, and graceful handling of transmission errors. Early versions of the system provided digital transmission of the video signal at 17 Mb/s with five-channel surround-sound CD-quality audio at 0.5 Mb/s.

In summary, the basic features of the HDTV system that will be implemented are as follows:

- higher spatial resolution, with an increase in spatial resolution by at least a factor of two in both the horizontal and vertical directions;
- higher temporal resolution, with an increase in temporal resolution by use of a progressive 60-Hz temporal rate;
- higher aspect ratio, with an increase in the aspect ratio to 16 : 9 from 4 : 3 for standard TV, providing a wider image;
- multichannel CD-quality surround sound with at least four to six channels of surround sound;
- reduced artifacts as compared to analog TV by removing the composite format artifacts as well as the interlace artifacts;
- bandwidth compression and channel coding to make better use of terrestrial spectrum using digital processing for efficient spectrum usage;
- interoperability with the evolving telecommunications and computing infrastructure through the use of digital compression and processing for ease of interworking.

F. Organization, Storage, and Retrieval Issues

Once multimedia material is compressed and coded, it needs to be sent over a network to one or more end users. To ensure that the communication does not break down in transmission, we need to ensure the method and speed of delivery of the material (via either a streaming implementation or a full download), we have to provide mechanisms for different resolutions of the multimedia material (depending on the capabilities of the receiving system), and we have to provide a guaranteed QoS so that the received multimedia signal has essentially the quality that is expected and is being paid for. In this section, we discuss each of these issues briefly.

1) Streaming Issues for Video: In Section III-C4 of this paper, we discussed the issues involved in streaming transmission of speech and audio over data networks. Much the same discussion holds for streaming of video material—perhaps the only difference is that video is much higher rate data than speech or audio, and a great deal of the time video requires only one-way streaming (movies, shows, documentaries, video clips, etc.) and can therefore tolerate long delays in the streaming network.

As discussed in Section III-C4, the four elements of a streaming system are:

- the compressed (coded) information content, e.g., audio, video, multimedia data, etc.;
- the server;
- the clients;
- the data network (e.g., the Internet) and the connections of the server and the clients to the data network.

A successful streaming application requires a well-designed system that takes into account each of these elements.

Currently, streaming in data networks is implemented as part of the application-layer protocols of the transmission, i.e., it uses UDP's and TCP at the transport layer. Because of the known shortcomings of TCP, most streaming implementations are based on the inherently unreliable UDP protocol. Thus, whenever there is network congestion, packets are dropped. Also, since delay can be large (on the order of seconds) and often unpredictable on the Internet, some packets may arrive after their nominal presentation time, effectively turning them into lost packets. The extent of the losses is a function of the network congestion, which is highly correlated with the time of day and the distance (in terms of the number of routers) between the client and the multimedia source. Thus, streaming itself does not inherently guarantee high-quality or low-delay playback of real-time multimedia material.

The practical techniques that have evolved for improving the performance of streaming-based real-time signal delivery can be classified into four broad areas, namely:

1) *client-side buffer management:* determining how much data needs to be buffered both prior to the start of the streaming playback and during the playback, and determining a strategy for changing the buffer size as a function of the network congestion and delay and the load on the media server;
2) *error-resilient transmission techniques:* increasing client-side resilience to packet losses through intelligent transport techniques, such as using higher priority for transmitting more important parts (headers, etc.) of a stream and/or establishing appropriate retransmission mechanisms (where possible);
3) *error-resilient coding techniques:* using source (and perhaps combined source and channel) coding tech-

niques that have built-in resilience to packet losses (e.g., the LPC methods discussed earlier for speech);

4) *media control mechanisms:* using efficient implementations of VCR-type controls when serving multiple clients.

None of these techniques is sufficient to guarantee high-quality streaming, but in combination they serve to reduce the problems to manageable levels for most practical systems.

2) Layering and Embedding of Images and Video: Throughout the discussion of the methods for coding and compression of speech, audio, image, and video signals, we have referred to and discussed layering and embedding or scalable methods of coding the compressed signal so that the transmission of the coded material could be matched to a range of receiver client capabilities. Hence, we showed how images could be sent in a progressive format so that low-resolution or low-quality versions could be first received and acted upon (at the client side) immediately, and higher resolution/higher quality versions of the images could be displayed if the client waited for a longer time before taking action or specifically requested the higher quality versions. In addition, if the client-side terminal was incapable of taking advantage of the full resolution or quality of the multimedia signal, it could signal the server or the network not to send all layers of the signal but only those that could be used at the client terminal, thereby lowering the transmission requirements without sacrificing quality at the client.

3) QoS Issues: The ultimate test of any multimedia system is whether it can deliver the QoS that is required by the user of the system. The ability to guarantee the QoS of any signal transmitted over the POTS network is one of the key strengths of that network. For reasons discussed throughout this paper, the packet network does not yet have the structure to guarantee QoS for real-time signals transmitted over the packet network. Using the standard data protocol of TCP/IP, the packet network can provide guaranteed eventual delivery for data (through the use of the retransmission protocol in TCP). However, for high-quality, real-time signals, there is almost no possibility of retransmission.

The ultimate solution to this problem lies in one of three directions.

1) Significantly increased bandwidth connections between all data servers and all clients, thereby making the issues involved with traffic on the network irrelevant. This solution is highly impractical and somewhat opposed to the entire spirit of sharing the packet network across a range of traffic sites so that it is efficiently and statistically multiplexed by a wide range of traffic.

2) Provide *virtual circuit* capability for different grades of traffic on the packet network. If real-time signals were able to specify a virtual circuit between the server and the client so that all subsequent packets of a specified type could use the virtual circuit without having to determine a new path for each packet, the time to determine the packet routing path would be reduced to essentially a table lookup instead of a complex calculation. This would reduce the routing delay significantly.

3) Provide *grades of service* for different types of packets, so that real-time packets would have a higher grade of service than a data packet. This would enable the highest grade of service packets (hopefully reserved for real-time traffic) to bypass the queue at each router and be moved with essentially no delay through routers and switches in the data network. MPEG-4 is implementing this approach.

There are a number of proposals to provide virtual circuits and grade of service, including the proposed RSVP protocol and the new IPv6 protocol for packet reservation and transmission services [38]. If these protocols are adopted, there are good prospects for guaranteed QoS to be available for multimedia sessions over packet networks.

G. Access to Multimedia

A key component of multimedia systems that are usable by real people is the capability of matching the user (the client) to the machine (the server) that provides the multimedia experience. Anyone who has tried to do Internet telephony or video teleconferencing or to retrieve combinations of video, text, audio, and data, over either the POTS or packet networks, already knows that the user interface for such systems is crucial to their successful use.

Multimedia systems have a natural, highly developed, graphical user interface when accessed via a standard PC. For multimedia systems to have their maximal impact, however, they need to be equally accessible from other devices, such as an ordinary telephone with only a touch-tone or voice interface, as they are from a PC. Of course, the user interface and the interaction will vary greatly in these different access modes, but the key concept is that the multimedia system essentially works properly, no matter what the access mode.

Perhaps the most popular user interface for multimedia systems is the GUI, which has evolved over the past decade and has become a highly effective tool for a vast range of applications. The basic design principles of GUI's are well understood [39] and consist of applications taking advantage of the following capabilities.

- *Continuous representation* of the objects and actions of interest. The goal is to keep the graphical objects (the things the system can do to exercise features of the application) "in the face" of the user so that it becomes both obvious and intuitive what can be done next and what will make things happen in a desired manner.
- *Rapid, incremental, and reversible* operations whose *impact* on the object of interest is *immediately* visible. Here, the goal is to make sure that every action that

the user can take has an immediate and unmistakable response that can easily be undone if the resulting state is incorrect or not the one that the user desires.

- *Physical actions,* or labeled button presses, instead of complex syntax with natural language text commands. Here, the point is that uniquely and well-labeled buttons, pull-down menus, etc. have a simple and clear interpretation that both the user and the system can understand, whereas alternative ways of requesting action, such as typing textual commands, invariably leads to ambiguity and confusion and should therefore be avoided

To some extent, all these principles reflect an understanding of the limitations of human cognitive processing. Humans are limited in their ability to both recall known information and retain large amounts of information in working memory. Continuous representation addresses the user's difficulty of remembering what options are available, while having physical actions associated with labeled button presses means that users do not have to remember details of a command syntax. Similarly, rapid operations with immediate, visible impact maintain the user's attention on the task at hand, facilitating efficient task completion.

To the extent that clients can use a GUI, the interface issues are well understood based on the above principles. For a wide range of applications, however, use of a GUI is not possible, e.g., the user experience is over an ordinary telephone connection or a terminal without a mouse and a keyboard or a touch screen. For such situations, it is essential that access to the multimedia signal be via a spoken natural language interface or via an agent interface. It is also essential that media translation tools be available to match the client capabilities, e.g., text-to-speech conversion to convert textual material to spoken form for presentation aurally to the client. We discuss the issues in creating spoken language interfaces to machines, in doing media conversion, and in providing agent interfaces in the following sections.

1) Spoken Language Interfaces: Although there exist a large number of modalities by which a human can have intelligent interactions with a machine, e.g., speech, text, graphical, touch screen, mouse, etc., it can be argued that speech is the most intuitive and most natural communicative modality for most of the user population. The argument for speech interfaces is further strengthened by the ubiquity of both the telephone and microphones attached to personal computers, which affords universal remote as well as direct access to intelligent services.

To maximize the benefits of using a speech interface to provide natural and intelligent interactions with a machine, the strengths and limitations of several technologies need to be fully understood. These technologies include:

- coding technologies that allow people to efficiently capture, store, transmit, and present high-quality speech and audio (we have already extensively discussed the coding technologies in Sections III-B and III-C);
- speech synthesis, speech recognition, and spoken language understanding technologies that provide machines with a mouth to converse (via text-to-speech synthesis), with ears to listen (via speech recognition), and with the ability to understand what is being said (via spoken language understanding);
- user interface technologies, which enable system designers to create habitable human–machine interfaces and dialogues that maintain natural and sustainable interactions with the machine.

The earliest spoken language interfaces in telecommunications services were interactive voice response (IVR) systems, where the system output was speech (either prerecorded and coded or generated synthetically) and user inputs were generally limited to touch-tone key presses [40]. Currently, IVR is used in many different application domains, such as electronic banking, accessing train-schedule information, and retrieval of voice messages. Early implementations of spoken language interfaces (SLI's) often simply replaced the key-press commands of IVR systems with single-word voice commands (e.g., push or say "one" for service 1). More advanced systems allowed the users to speak the service name associated with the key push (e.g., "For help, push 'one' or say 'help'"). As the complexity of the task associated with the IVR applications increased, the systems tended to become confusing and cumbersome, and the need for a more flexible and more habitable user interface became obvious to most system developers. This is especially the case for multimedia systems, where the integration of the different media often lead to complex interactions with the machine.

In the next section, we provide a brief overview of the media conversion technologies associated with SLI's, namely, speech synthesis for conversion of text into spoken messages, speech recognition for conversion of spoken inputs into text, and speech understanding for conversion of spoken inputs into requests for action on the part of the machine. In the following section, we show how we use the imperfect technologies of synthesis, recognition, and understanding to provide practical SLI systems. In particular, we concentrate on the way in which these technologies are used in voice-controlled intelligent-agent interfaces to machines.

2) Media Conversion:

a) Speech synthesis: SLI's rely on speech synthesis, or TTS systems, to provide a broad range of capability for having a machine speak information to a user [41], [42]. While for some applications, it is possible to concatenate and play prerecorded segments, this is not possible in general. Many applications are based on dynamic underlying information sources for which it is difficult to predict what the system will need to say. These include applications that involve generating natural language sentences from structured information, such as a data base record, as well as those involving unstructured information, such as e-mail messages or the contents of a Web page. These applications can also take advantage of the MPEG-4 face animation

Table 5 Word Error Rates for Speech Recognition and Natural Language Understanding Tasks (Courtesy of J. Makhoul, BBN)

CORPUS	TYPE	VOCABULARY SIZE	WORD ERROR RATE
Connected Digit Strings	Spontaneous	10	0.3%
Airline Travel Information	Spontaneous	2500	2.0%
Wall Street Journal	Read Text	64,000	8.0%
Radio (Marketplace)	Mixed	64,000	27%
Switchboard	Conversational Telephone	10,000	38%
Call Home	Conversational Telephone	10,000	50%

capability of Fig. 31 to provide visual speech synthesis capability via talking heads.

TTS systems are evaluated along two dimensions, namely, the intelligibility of the resulting speech and the naturalness of the speech. Although system performance varies greatly for different tasks and different synthesizers, the best TTS systems achieve word intelligibility scores of close to 97% (natural speech achieves 99% scores); hence, the intelligibility of the best TTS systems approaches that of natural speech. Naturalness scores for the best TTS systems, as measured by conventional MOS scores (see Section III-B2) are in the 3.0–3.5 range, indicating that the current quality of TTS is judged in the fair-to-good range, but most TTS systems still do not match the quality and prosody of natural speech.

b) Speech recognition and spoken language understanding: The ultimate goal of speech recognition is to enable a machine literally to be able to transcribe spoken inputs into individual words, while the goal of spoken language understanding research is to extract meaning from whatever was recognized [43], [44]. The various SLI applications discussed earlier have widely differing requirements for speech recognition and spoken language understanding; hence, there is a range of different performance measures on the various systems that reflect both the task constraints and the application requirements.

Some SLI applications require a speech recognizer to do word-for-word transcription. For example, sending a textual response to an e-mail message requires capabilities for voice dictation, and entering stock information or ordering from a catalogue may require entering number sequences or lists of data. For these types of systems, *word error rate* is an excellent measure of how well the speech recognizer produces a word-for-word transcription of the user's utterance. The current capabilities in speech recognition and natural language understanding, in terms of word error rates, are summarized in Table 5. It can be seen that performance is very good for constrained tasks (e.g., digit strings, travel reservations) but that the word error rate increases rapidly for unconstrained conversational speech. Although methods of adaptation can improve performance by as much as a factor of two, this is still inadequate performance for use in many interesting tasks.

For some applications, complete word-for-word speech recognition is not required; instead, tasks can be accomplished successfully even if the machine only detects certain key words or phrases within the speech. For such systems, the job of the machine is to categorize the user's utterance into one of a relatively small set of categories; the category identified is then mapped to an appropriate action or response [45]. An example of this type of system is AT&T's "How May I Help You" (HMIHY) task, in which the goal is to classify the user's natural language spoken input (the reason for calling the agent) into one of 15 possible categories, such as billing credit, collect call, etc. Once this initial classification is done, the system transfers the caller to a category-specific subsystem, either another artificial agent or a human operator [46]. *Concept accuracy* is a more appropriate measure of performance for this class of tasks than word accuracy. In the HMIHY task, word accuracy is only about 50%, but concept accuracy approaches 87%.

Another set of applications of speech recognition technology is the so-called spoken language understanding systems, where the user is unconstrained in terms of what can be spoken and in what manner but is highly constrained in terms of the context in which the machine is queried. Examples of this type of application include AT&T's CHRONUS system for air-travel information [47] and a number of prototype railway information systems. As in the HMIHY example, results reported by Bennacef *et al.* [48] show speech-understanding error rates of 6–10% despite recognition error rates of 20–23%. These results demonstrate how a powerful language model can achieve high understanding performance despite imperfect automatic speech recognition (ASR) technology.

3) Agent Interfaces: In this section, we discuss the design principles of SLI's to machines and use the agent interface as the example of how such interfaces are implemented and how the resulting human–machine dialogues are designed to best match both human and machine characteristics.

Much as was the case for well-designed GUI's, well-designed SLI's also address human cognitive limitations. However, the nonpersistent, temporal nature of audio signals, coupled with the limited capacity of auditory memory, impose additional requirements for SLI design. Obviously, without a visual display, different methods are necessary to instantiate design principles like "continuous representation" and "immediate impact." Furthermore, whereas a persistent visual display permits presentation of large amounts of information in tabular format, which can be easily scanned and browsed, audio-only interfaces must summarize and aggregate information into manageable pieces that a user can process and cope with effectively.

For example, consider a complex task like information retrieval, where the amount of information to be provided to the user is large. In this case, the limitations on human auditory processing capacity impose constraints on the system that drive SLI's toward dialog-based interactions. In these dialogs, the system presents information incrementally, and multiple exchanges between the agent and the user are often necessary to extract the specific information the user requires.

Thus, in addition to SLI design principles covering a single exchange between the user and the agent, SLI design extends to the broader problem of dialogue management [49]. For example, dialogue management requires that the system keep track of the current context, including what the user has already said, what the system has already provided, and previous misrecognitions and misunderstandings by the system.

The remainder of this section presents several examples of the application of the principles of continuous representation, immediate impact, reversibility, incrementality, and summarization/aggregation in audio-only speech-enabled interfaces.

Simulating a "continuous representation" in an audio-only interface requires a tradeoff between providing a full representation of the available options in each audio prompt and not providing sufficient information. Novice users might desire (and require) a full representation of all the options, but this is likely to be unacceptably long for an experienced user. One compromise strategy is to provide prompts in a "question—pause—options" format, where the "options" serve as a reminder of what the user can say at this point in the dialogue. The next section of text shows an example of this strategy, in the context of an anthropomorphic personal agent metaphor, which can be used as an interface to a suite of call control, messaging, and multimedia information retrieval services [50]:

> *Agent:* Maxwell here, what can I do for you?
> *User:* (says nothing within allocated two-second speech interval).
> *Agent:* You can say "call," followed by a name or number, "Get my messages," or "Get me the employee directory." For more options, say "Help me out."
> *Design principle:* continuous representation.

In this example, the user is reminded of the available options when the agent detects that the dialogue is not progressing as expected. That is, because the system has not detected speech within two seconds after the agent's initial prompt ("What can I do for you?"), the assumption is that the user may not know what to do. At that point, the agent explains how to perform the most frequently used functions, as well as how to find out about more options.

In addition to reminders when a time-out has occurred, another strategy for establishing a "continuous representation" in an audio-only interface is to provide consistency across features in a multifeatured application (e.g., the personal agent for communications services). This consistency can be achieved by using a vocabulary that is "almost always available" and serves the same function regardless of where the user is in the feature space. For example, at any point in the interaction, the command "help me out" obtains context-dependent instructions, and the command "cancel" undoes the preceding request, as shown in the text below. In this example, the user gives the command "help me out" first at the top level of the quasi-hierarchical feature space and then later inside a module that provides access to an employee directory. In each case, the user receives a help message from the agent appropriate to the options available at that point in the interaction.

> *Agent:* Maxwell here. What can I do for you?
> *User:* Help me out.
> *Agent:* You can say "call," followed by a name or number, "Get my messages," or "Get me the employee directory." You can always say "I'm done . . .
> *User (barging in over Agent's speech):* Get me the employee directory.
> *Agent:* First message . . .
> *User (barging in over Agent's speech):* Cancel.
> *Agent:* Maxwell here.
> *User:* Get me the employee directory.
> *Agent:* Employee directory.
> *User:* Help me out.
> *Agent:* Say the first and last name of the person you want to call.
> *Design principle:* continuous representations and immediate impact.

A third strategy for providing continuous representation in an audio-only interface is to give users audible cues to orient them with the task space. These cues can be brief, distinctive sound effects ("earcons," by analogy with "icons" in the GUI) or spoken prompts that the agent says to indicate the user's "location." In the example shown in the text above, the prompt "Maxwell here" cues the user that the top-level vocabulary is active, and the prompt "employee directory" indicates that the interaction has entered the employee directory module.

The agent's response of "employee directory" also satisfies the design principle of providing immediate, observable impact by providing implicit feedback that the user's previous request was understood and acted on correctly. In contrast, the user's first request of "Get me the employee directory" in the text above was misrecognized by the system as "Get my messages." In this case, the agent's implicit confirmation statement "First message" was an immediate cue to the user that an error had occurred. The user then spoke the command "Cancel," which caused the system to undo the previous action (the erroneous foray into the messaging module) and return to the top level of the interaction. To achieve the design principle of immediacy, the system must have real-time, low-latency responses and must also allow the user to "barge in" while the system is speaking [51]. Effective audio interfaces must be able to listen even while they are presenting prompts and must stop speaking when they are interrupted by the user (following human conversational etiquette). This principle provides a sense of continuity and natural flow to the dialogue.

The principle of incrementality is illustrated in the interaction in the text below, taken from an Air Travel Information System application. In this case, the user has provided the originating airport but has not given sufficient information for the agent to disambiguate the destination airport, so the agent asks a clarifying question to determine

the destination and then proceeds to ask about the next piece of information that the system needs in order to complete the query to the air-travel data base.

User: I want to go from Boston to San Francisco.
Agent: San Francisco is served by two airports, San Francisco International and Oakland International. Which airport do you want?
User: San Francisco International.
Agent: When do you want to travel from Boston to San Francisco International Airport?
Design principle: incrementality I.

Incrementality is also a useful design strategy for handling instances when the speech recognizer only recognizes a portion of the utterance and needs to reprompt for the missing information, as in the call completion example in the text below.

User: Call John Doe.
Agent: Call who?
User: John Doe.
Agent: Calling John Doe.
Design principle: incrementality II.

The principle of summarization and aggregation of information is particularly important to audio-only interfaces, where the options for scanning and browsing information are quite limited. An example of the use of summarization and aggregation from a system that provides voice access to electronic mail is shown in the text below. In this example, rather than reading out a list of 20 electronic-mail message subjects and senders, the system tells the user how many messages there are and provides some additional information that has been previously specified as of importance to this user (i.e., "important messages from President Clinton and from Vice President Gore").

Agent: You have 20 new e-mail messages, with important messages from President Clinton and from Vice President Gore.
User: Do I have any messages about meetings for today?
Agent: You have three messages about meetings for today: one from Gore about "NII meeting at 3 p.m.," one from Shalala about "budget meeting" and one from your secretary about "staff meeting at 8 a.m."
User: Please read the messages about meetings.
Design principle: summarization and aggregation.

In addition to these general design principles, successful spoken dialogue interactions may be facilitated by customizing and adapting the system to the needs and preferences of individual users. To customize the interaction effectively, an accurate user model is required. The user model includes assumptions about what the user knows about the system and how the user prefers to interact with the system.

For example, for the personal communication agent application, the user model for an "expert" user might include the following:

a) the expert knows and remembers what the system feature set is and what commands invoke those features;
b) the expert prefers terse, implicit confirmation strategies that move the dialogue along as quickly as possible;
c) the expert typically speaks to the system in terse telegraphic commands.

In contrast, the user model for a "novice" user might be that:

a) the novice remembers a few commands but often will need reminders of what is available;
b) the novice may prefer more thorough confirmation to assure him that the dialogue is progressing correctly;
c) the novice is apt to provide only partial information, requiring more frequent use of incremental strategies.

Based on these differences in underlying assumptions about expert and novice users, the customized dialogues shown below might be proposed for the two user populations. As a user gains experience with the system, the system could track the user's behaviors, and after the number of successful uses of a feature exceeds some critical threshold, the agent could offer the user the option of shifting from the novice interaction to the expert version of the system.

Agent: Annie here.
Expert: Get my labels.
Agent: Labels.
Expert: Delete the label for Candy Kamm.
Agent: Deleting Candy Kamm. Labels, what next?
(Expert dialogue)
Agent: Annie here. What would you like me to do?
Novice: Help.
Agent: Here's what I can do for you: To make a call say "Call" followed by a name or number. To administer your list of names say "Get my labels." To get your messages say ...
Novice (interrupting): Get my labels.
Agent: I've got your labels. What would you like me to do?
Novice: Delete a label.
Agent: Say the name or number you want to delete.
Novice: Candy Kamm.
Agent: Are you sure you want to delete Candy Kamm?
Novice: Yes.
Agent: OK, I'll delete Candy Kamm and I'll get right back to you.
(Novice dialogue)

User-specific usage patterns and preferences for system features offer an additional information source for customizing and adapting the interface to these systems. For example, if the personal agent detects that the user has called a particular telephone number repeatedly over a short time period, the agent may offer the user the option of

adding that number to the user's personal voice dialing list. The impact of customization and system adaptation on task performance and user satisfaction with complex human–computer interactions has not been explored systematically and is one of many aspects of SLI's that need further study.

H. Searching of Multimedia Documents

The ability to represent multimedia documents, consisting of combinations of speech, audio, image, video, handwriting, and text, in digital form has enabled both the storage and processing of these media by digital computers. Consequently, plain text data bases have evolved to store multimedia information that can be accessed, browsed, sorted, reorganized, indexed, and searched. As large volumes of distributed multimedia information become both commonplace and in great demand, the need to manage this information efficiently and rapidly becomes essential. An important issue in effective utilization of multimedia information repositories is the selective retrieval of information for searching and browsing applications. In this section, we discuss the mechanisms for searching of multimedia documents by text, speech, audio, image, video, and collaborative indexing methods. In the next section, we discuss the mechanisms for browsing of multimedia documents.

It has been said that the value of storing any material can be measured by the amount of effort it takes to find and retrieve the material when it is needed for some purpose. For textual material, highly sophisticated methods have evolved over the past few decades, based on both direct text matching techniques and associative matching based on semantic interpretations of the requested text terms for the match. For other multimedia modalities, such as speech, audio, image, video, etc., such advanced matching techniques are neither well known nor well developed.

Intelligent retrieval of multimedia information calls for an effective indexing method. The heterogeneous nature of multimedia information requires the use of a range of techniques from several fields to achieve this goal. In some cases, these techniques are applied in isolation to an individual component of the multimedia stream. In other cases, two or more of the techniques may be applied together in a collaborative way.

A complete discussion of the techniques in each of these fields is outside the scope of this paper. The next sections are aimed at giving the reader a general idea of the possible approaches to indexing the media streams that comprise digital multimedia libraries and data bases.

1) Text-Based Indexing: Among the media that comprise multimedia information sources, text was the first medium to be stored and indexed in digital form. There is a large body of knowledge on text-based information retrieval. The current state of the art for indexing is far more advanced for text compared to other media in a multimedia presentation. Repositories of textual information can exist in two forms, namely, *structured* and *unstructured.* Structured text repositories are those that have been organized into a special form consisting of predefined fields, such as data bases. This type of textual information lends itself easily to traditional data base queries that are based on the information in specific data fields. A more challenging task is to index unstructured text, also known as natural language, such as that found in printed or electronic publications. This indexing may be geared toward populating a standard data base or for use with more general information retrieval tasks. Unstructured text documents can be selectively retrieved by *full-text search.* This is based on creating an *inverted index* that, for each word, records the documents (and the locations within each document) in which it appears. Full-text searches can be performed using multiple words. Additional constraints are imposed using Boolean logical operators. Special characters are used to find partial matches. Morphological normalizations such as *expanded recall* (or word stemming) are used to find possible variants of a given word. Semantic normalizations are used to search for classes of words with similar meanings. Proximity constraints are also applied to retrieve only those documents in which certain words appear close to each other. Most of the search engines available on the World Wide Web (e.g., Lycos, AltaVista, etc.) are based on a full-text indexing scheme. A combination of field-based and full-text indexing can also be used.

The use of exact matches to generate an inverted index results in missing relevant information in the presence of spelling errors. This situation arises often due to errors in manual text entry or when the text is generated automatically from speech using speech-to-text (machine recognition of speech) conversion or from printed text using OCR methods—both of which are error-prone processes. This situation is remedied by using *approximate matching* techniques that establish the similarity between words that do not match exactly.

The techniques discussed above are directed at retrieving all or part of a textual document from a large document data base. Another interesting and useful task is that of extracting information from large textual data bases, namely, *information extraction* [52]. The goal here is to retrieve relevant documents selected by a document-retrieval method, extract relevant fragments from the retrieved documents, derive useful information from these fragments, and piece them all together into a useful document that satisfies the information query.

Another area of interest in text-based information indexing is that of text decomposition into *text segments* and *text themes* [53]. Text segmentation involves the identification of contiguous pieces of text that exhibit internal consistency and are distinguishable from surrounding text segments. Text theme decomposition is the process that decomposes the text based on its semantic content. These decompositions and the interactions between them are used for text summarization and the segmentation of multitopic documents into individual topics.

2) Speech Indexing: If a multimedia document contains a speech stream, then this stream can be used to index the document in a number of ways, ranging from full

recognition of the speech material to event detection within the speech stream.

Conversion from speech-to-text or, more appropriately, recognition of the speech stream can be used to generate transcriptions of the spoken material. Text-based indexing can then be used to index the speech based on the resulting text transcription. Large vocabulary (LV)ASR's (e.g., WATSON from AT&T, SPHINX from Carnegie-Mellon University, etc.), which require sophisticated acoustic and language models and are task-domain specific, can be used for this task. With the appropriate acoustic and language models, the generated transcripts can have word error rates from as little as 5% to 50% or more [54]. While this may severely limit the readability of the automatically generated transcripts, the resulting transcripts can nevertheless be used with acceptable levels of success by text-based indexing information-retrieval algorithms [55]. The application of LVASR to domains to which the existing language and acoustic models do not apply results in much higher word error rates.

An alternative to LVASR is to detect or spot in the speech stream certain words and phrases in the unconstrained speech. This process is known as *word spotting.* This approach uses a large corpus of labeled speech data to estimate the parameters of a set of *hidden Markov models* (HMM's) [56], which are statistical representations of speech events such as the words and phrases to be spotted as well as general speech models and background models. Given an unknown speech stream, the best matching set of HMM's is then used to spot the desired words and phrases by an efficient searching and matching algorithm [57].

Speech indexing can also be achieved by performing acoustic indexing using phone lattices [58]. This approach is similar to word spotting. It uses a set of HMM's to generate a number of likely phone sequences by off-line processing. The computational requirements for the off-line processing are comparable to that of LVASR. This lattice can be searched rapidly to find phone strings comprising a given word. The rapid search comes at the expense of storage for the lattice, which requires about a megabyte per minute of speech. This approach does not require a language model to be built. However, acoustic models need to be constructed for the individual phones in order for the method to work. This approach has been successfully employed in a video mail-retrieval system to retrieve relevant messages by speech queries.

Depending on the application, an acceptable level of speech indexing can be achieved without the explicit recognition of the spoken words by detecting *speech events*. These are events such as transitions between speech and silence and transitions between two speakers. A higher level of indexing would involve speaker identification. Such information can be utilized to align the speech component of a multimedia presentation with available transcripts to allow users to retrieve selected speech segments. The search, in this case, is entirely based on the textual information, and the speech segments are retrieved based on their one-to-one correspondence with the text segments. Speaker identification has been successfully used to align the audio recordings and the transcripts of the proceedings of the U.S. House of Representatives [59]. The transcripts carry time stamps and information about the speakers. A text parser extracts this information. Acoustic models constructed for each speaker are then used to locate speaker transition points in the recorded speech. A set of Gaussian models (one per each speaker) based on cepstral features are computed and are used in conjunction with the timing information and speaker sequence information to perform a Viterbi alignment of the transition points. Once the speech is segmented and associated with the transcript, it can be selectively retrieved by text-based queries.

3) Audio Indexing: Just as we can use the speech stream in a multimedia document to either generate a textual representation or find user-specified events in the speech (e.g., words, phrases, speaker occurrences, etc.), we can also use an audio stream to automatically detect and segment the audio stream into significant events (e.g., presence or absence of audio, presence of specific musical signatures for instruments, combinations of instruments, etc.), as well as the presence of singing or voice (along with the appropriate transcription of the words) and even the creation of the musical score from which the audio was created. Such events can be used to index the material so that rapid access to and searching of the music can be accomplished. The methods used for audio indexing of multimedia content are beyond the scope of this paper.

4) Image Indexing: The human visual system is extremely good at determining the content of images. This task, however, poses real problems for machines [60]. Ideally, we would like a machine to be able to examine an image (or eventually a video sequence of images) and automatically classify the objects in the image, the relationship of the objects to each other, the theme of the image (e.g., still life, sporting event, indoor scene, outdoor scene, action scene, etc.), the sharpness and intensity of the image, and even potentially the emotions of the people shown in the image. Practically speaking, automatic techniques for image understanding and classification are still in their infancy and at best are able to classify images according to color and brightness characteristics (generally based on histograms of color and intensity), texture characteristics (generally based on foreground and background color maps), extracted shapes in the image (presence or absence of predefined objects such as blocks, people, toys, etc.), and associations with external characterizations of the image, such as figure captions or textual descriptions in the body of an accompanying text document.

Short of having automatic methods for classifying images based on the objects detected within the image, it is desirable to be able to match image data bases to token examples of desired matching images based on different matching criteria, such as shape matches, color matches, texture matches, etc. A great deal of work has gone on to determine how well such "proximity matches" can be made from large generalized image data bases with more

than 20 000 images. Generally, the results of such matching procedures are a list of the N best matches (typically $N = 100$), with the user asked to choose one of the N matches based on visual selection. This technique tends to mediate computer errors with human choice for some types of images [61].

5) Video Indexing: Video indexing can also be achieved by utilizing data base information about the content and structure of the video material, by detecting scene fades and scene breaks, by matching perceived camera motions, and by utilizing as much side information as available from things like closed-captioned text that accompanies the video or from text created from the speech stream associated with the video stream. The indexing of video material by such content-based sampling methods is the impetus behind a system called Pictorial Transcripts, which will be described later in this paper.

I. Browsing of Multimedia Documents

There are several reasons why powerful browsing capabilities are essential in a multimedia information-retrieval system. Perhaps the most important reason is the one discussed in the previous section, namely, that the ultimate multimedia indexing system would require solving three of the most challenging problems in computer science. Document indexing requires zero-error OCR for all document types, fonts, character sizes, special symbols, and languages. Audio and speech indexing requires perfect speech understanding (and audio understanding) in all acoustic environments, for all speakers, and for all languages. Image and video indexing requires locating and identifying all objects (and ultimately their relationships) in complex scene images, a problem commonly referred to as the "general vision problem." Hence, for the foreseeable future, the only way to access and search a large range of multimedia documents is via a combination of indexed searching and browsing.

The limited precision of multimedia search mechanisms often results in the extraction of large collections of documents from the library, some of which are not of interest to the user. This leaves the task of filtering the information to the user. This process of searching is called browsing. Even in the case where a preliminary search provided a relevant document, the information of interest may be distributed at different parts of the document. In this case, browsing mechanisms are needed to allow the user quickly to navigate though the document and bypass irrelevant information. Other useful browsing capabilities to get around the limitations of linear media (e.g., video and audio) are speedup mechanisms such as pitch preserving fast (or slow) replay of audio, speed increase and decrease for video replay, and audio and video skims. Tools such as hypertext markup language (HTML) and the user interfaces provided by the associated viewing software have provided powerful mechanisms for navigating through heterogeneous data.

As anyone who has browsed a standard book with pictures, charts, tables, graphs, etc. knows, there are several ways of browsing a book to find material of interest. These include the following.

- Image-based browsing, where a stream of images is shown (with appropriate links to the referenced text, speech, audio, and video streams). The user browses the image sequence until an image of interest in found, at which point the additional media are used to determine the value of the found material.
- Scene-based browsing, where a stream of video breaks is shown (typically a single image per video break). The user browses the scene breaks until a scene of interest is found, at which point the video is played completely (either forward or backward) from that point on. This type of browsing is useful for finding a particular place in a video sequence or for editing video pieces in production of a new video document.
- Video-skimming browsing, where the video is played at a rate significantly faster than real time (but at a fixed number of frames/second) and the user browses until a section of interest is found. This mode is similar to the VCR-like controls used in the MPEG video standard, as well as to the VCR controls used in home systems for fast forwarding to a section of interest in a videotape.
- Audio-skimming browsing, where the audio is played at a rate significantly faster than real time and the user browses until a section of interest is found. This mode is similar to high-speed audio playback, but in order to maintain intelligibility in the high-speed mode, the signal must be processed to avoid the "Donald Duck" effect of increases in the pitch of sounds due to the high-speed playback mode.

The Pictorial Transcripts system, as described in the next section, illustrates the key concepts of browsing.

J. Networking, Multiplex, and Control

A final set of technologies that are intimately related to multimedia communications are networking, multiplexing, and control. By networking, we refer to the type of network that is used for the transport of the multimedia information. By control, we refer to the protocols that are used to set up, confirm, alter, and tear down a multimedia connection. By multiplexing, we refer to the protocols that are used to combine and break down the different multimedia data streams that are transmitted through the network. Each of these technologies has been referenced in earlier parts of this paper. We talked about the various types of networks in the introductory sections, and we looked at control and multiplexing when discussing the MPEG standards, especially MPEG-2 and MPEG-4. In this section, we provide a summary of the key aspects of each of these technologies.

There are a number of different types of networks that can be used to transport multimedia data. The first network that comes to mind is the POTS or public switched telephone network that has carried the voice traffic for telecommunications for the past 100 years. The POTS network includes both analog and digital links, with ISDN being the most

common digital transmission system that can be accessed at the customer premises site. The main virtue of the POTS network is that it provides a guaranteed QoS since the switched channel is dedicated to a single user and provides an end-to-end connection at a constant bit rate. Although the basic channel bit rate of the POTS network, 64 kb/s, is rather low, several channels can be combined to provide higher data rate capability (this is essentially how ISDN provides 128-kb/s channels to the home). The basic channel has very low time jitter and a relatively low transmission delay (strictly based on the length of the connection and the speed of electromagnetic waves).

The alternatives to circuit-switched telephony are based on packet transmission over data networks. In this case, the user shares a large channel with all other uses in a multiplexed sense. The three most popular data-transmission protocols are TCP/IP, ATM, and frame relay. ATM was designed to handle both real-time (synchronous) and nonreal-time (asynchronous) traffic with low delay. Hence, the ATM packet size is 48 bytes for data (or speech or video) with 5 bytes for all header information. ATM routing of packets is done using virtual circuits that are set up using separate signaling packets. Both frame relay and IP packets are much larger than ATM packets. The larger packet size provides higher data efficiencies but more latency (delay) for real-time traffic in the network.

Packetized signals can be routed or switched. A router reads the packet header information and determines the destination and other packet routing information. The router must determine how to send the packet to the next destination (either another router or a switch). The priority that the router gives to the individual packets depends on information contained in the packet header. A packet switch uses a signaling packet to set up a virtual circuit for all packets associated with some stream and routes all subsequent packets along this virtual circuit. Hence, the relatively long time that routers need to decode packet headers is avoided by keeping a routing table for all virtual circuits in a cache. Packet switching helps reduce both the absolute delay and the jitter that are inherent in a router-based network. To gain the maximal advantage of packet switches, however, the entire end-to-end connection must be based on packet switches rather than on routers.

Many multimedia network connections actually use both the POTS and packet networks. For example, most consumers access the Internet through a dial-up telephone modem connected to a packet network via an ISP. In such a connection, the limitations on data rate, delay, and QoS are the aggregate of the limitations of each of the networks involved.

Multiplexing protocols are necessary to combine the different types of multimedia data streams into a single stream, either for transmission or storage. On a dedicated channel (e.g., a 64-kb/s POTS channel) the multiplexing scheme can use all of the capacity of the channel. On a packet channel, the multiplexing scheme can only use a portion of the channel (since it is shared with other users). We have already illustrated the use of multiplexing protocols in Sections III-E3 and III-E4 on MPEG coding, and will refer back to this concept in Section IV-A, when we discuss a multimedia conferencing system.

Control protocols are concerned with how two or more multimedia terminals exchange information about capabilities, how they issue commands to each other to configure themselves in various ways, and how they respond to each other as to status, etc. The concept here is that in order to maintain a viable multimedia link between two or more user locations, a handshaking process needs to be established. The purpose of this handshaking process is to synchronize all the user terminals about the types of signals that can be transmitted and received by the individual terminals, e.g., what types of speech, audio, image, and video coders can be utilized in the transmission, what speed network can be used to transmit the multimedia signals, etc. By way of example, one user terminal might want to let the other user terminals know that it has muted its microphone so that no speech will be transmitted during the muting interval. The ITU control protocols (to be discussed in Section IV-A) cover the basic procedures for call setup and operation, terminal capability exchange, mode initialization, methods for opening channels, and other general terminal procedures.

IV. Examples of Multimedia Systems

In this section, we discuss a number of multimedia processing systems that have been investigated at AT&T Labs, which show how the various technologies discussed in Section III come together in meaningful ways to create interesting and valuable multimedia applications. In particular, we discuss the following four applications:

1) multimedia conferencing systems, which primarily utilize speech-coding standards, video-coding standards, network transmission, stream multiplexing standards, and control standards as a series of piece parts to create a broad range of audio and audio/video teleconferencing capabilities;
2) the FusionNet video server, which combines packet control and search capability with POTS (over ISDN) reliability and guaranteed QoS to provide immediate and reliable access to audio/video programming streams over the Internet;
3) the CYBRARY digital-library project, which attempts to create a cyberlibrary that provides a user presence that is perceived to be "better than being there live" in terms of access to, retrieval, and viewing of full-scale color documents in their original form, color, and texture;
4) the Pictorial Transcripts system, which provides a compact representation of a full-motion video program in terms of content-sampled images, the associated audio stream, and a derived text stream, along with searching and browsing mechanisms for random access, retrieval, and playback (of either the audio portion or the audio-visual portion) of an arbitrary portion of the video program.

Table 6 Compression Results for Ten Selected Images. Column "Raw" Reports the Size in Kbytes of an Uncompressed TIFF File at 24 b/pixel and 300 pixel/in. Column "DJVU" Reports the Total Size in Kbytes of the DJVU-Encoded File. Column "Ratio" Gives the Resulting Compression Ratio of the DJVU Encoding. Column "JPEG-300" Reports the Size in Kbytes of 300 pixels/in, JPEG-Encoded Images with Quality Factor 25%. For the Sake of Comparison, Column JPEG-100 Reports the Size in Kbytes of 100 pixels/in JPEG-Encoded Images with Quality Factor 30%; However, the Quality of Those Images Is Unacceptable

File	Description	Raw	DJVU	Ratio	JPEG-300	/100
metric.tif	(Text on various backgrounds)	20,000	56	350	496	89
hobby002.tif	(Mail order catalog page)	24,000	80	300	412	82
plugin.tif	(Weekly magazine page)	21,000	66	318	527	102
pharm1.tif	(XVIIe century book page)	6,000	88	181	630	85
encyc2347.tif	(Dictionary page)	23,000	98	234	950	150
amend.tif	(US first amendment)	61,000	212	287	456	78
carte.tif	(XVIIIe century map)	32,000	146	220	586	100
wpost2.tif	(Newspaper page section)	21,000	96	218	435	64
curry239.tif	(Book page with picture)	13,000	76	171	490	53
curry242.tif	(Book page without picture)	14,000	40	350	410	73

A. Multimedia Conferencing Systems

Multimedia conferencing systems have been used for a number of years to provide an environment for shared meetings and shared workspaces that are spatially separated. With the advent of and the adherence to well-known standards for compressing and coding the signal components of the multimedia transmission (i.e., the audio and video signals), and with the increased availability of sufficient bandwidth to adequately display a simple conference-room video signal, the associated audio signal, and associated data signals, the popularity of multimedia conferencing systems has been steadily growing.

In recent years, the ITU has produced a number of international standards for real-time digital multimedia communications, including video and data conferencing. These H.32x standards specify the network for transmission, the video-coding standard, the audio-coding standard, the multiplexing standard, and the control standard. The most popular and widely used of these standards are the H.320 standard (created in 1990) for ISDN conferencing, the H.324 standard (created in 1995) for POTS conferencing, and the H.323 standard (created in 1996) for LAN/intranet conferencing. The basic components of each of these standards are shown in Table 6.

The H.320 standard is essentially the $p \times 64$ video conferencing system because of its deployment over ISDN channels, which come in bandwidth increments of 64 kb/s for each B-channel that is used. The H.320 standard supports real-time conversational two-way video and audio (one channel each) with provisions for optional data channels. Extensions allow multipoint operation, encryption, remote control of far-end cameras, document sharing, and broadcast applications.

The H.221 multiplex standard associated with H.320 conferencing mixes the audio, video, data, and control information streams into a single bit stream and uses synchronous time-division multiplexing with 10-ms frames. The H.242 control standard issues control commands and indications and does capabilities exchange. It operates over a fixed 400-b/s channel in the H.221 multiplex stream.

The success of the H.320 standard sparked the development of many extensions, including the H.323 standard for packet networks and the H.324 standard for low-bit-rate, circuit-switched networks. The major features of the H.323/H.324 standards include both improvements to the control and setup and features that support multimedia extensions, including:

- faster call startup upon initial connection;
- support for multiple channels of video, audio, and data;
- simpler and more flexible assignment of bandwidth among the various channels;
- separate transmit and receive capabilities;
- means to express dependencies between capabilities;
- explicit description of mode symmetry requirements;
- receiver-driven mode request mechanism;
- improved audio and video coding;
- large range of video modes and resolutions;
- cleaner mechanisms for extension to future standard and nonstandard features.

The new H.245 control protocol standard is based on "logical channels" with independent unidirectional bit streams with defined content, identified by unique numbers arbitrarily chosen by the transmitter, with up to 65 535 logical channels.

The H.324 standard is a "toolkit" standard, which allows implementers to choose the elements needed in a given application. An H.324 conferencing system can support real-time video, audio, and data or any combination. The mandatory components of an H.324 system include a V.34 modem, the H.223 multiplexer, the G.723.1 voice coder, and the H.245 control protocol. Video, audio, and data streams are all optional, and several of each kind may be used. H.324 enables a wide variety of interoperable terminal devices, including PC-based multimedia video-conferencing systems, inexpensive voice/data modems, encrypted telephones, World Wide Web browsers with live video, remote security cameras, and stand-alone video phones.

B. FusionNet Video Server [62]

A key problem in the delivery of "on-demand" multimedia communications over the Internet is that the Internet today cannot guarantee the quality of real-time signals such as speech, audio, and video because of lost and delayed packets due to congestion and traffic on the Internet. The FusionNet service overcomes this problem by using the Internet to browse (in order to find the multimedia material that is desired), to request the video and audio, and to control the signal delivery (e.g., via VCR-like controls). FusionNet uses either POTS or ISDN to actually deliver guaranteed QoS for real-time transmission of audio and video.

FusionNet service can be provided over a single ISDN B-channel. This is possible because the ISP provides ISDN access equipment that seamlessly merges the guaranteed QoS audio/video signal with normal Internet traffic to and from the user via point-to-point protocol (PPP) over dialed-up ISDN connections. Unless the traffic at the local ISP is very high, this method provides high-quality FusionNet service with a single ISDN B-channel. Of course, additional B-channels can always be merged together for higher quality service.

1) FusionNet System: The increases in the computing power and the network access bandwidth available to the general public on their desktop and home computers have resulted in a proliferation of applications using multimedia data delivery over the Internet. Early applications that were based on a first-download-then-play approach have been replaced by *streaming* applications [63], which start the playback after a short, initial segment of the multimedia data gets buffered at the user's computer. The success of these applications, however, is self-limiting. As they get better, more people try to use them, thereby increasing the congestion on the Internet, which, in turn, degrades the performance of such real-time applications in the form of lost and delayed packets.

Although several mechanisms, such as smart buffer management and the use of error-resilient coding techniques, can be employed at the end points to reduce the effects of these packet losses in a streaming application, these can only be effective below a certain packet-loss level. For example, generally, the amount of data to be buffered at the beginning of the playback and, hence, the start-up delay is determined in real time based on the reception rate and the rate used to encode the particular material being played. If the congestion level is too high, the amount of data to be buffered initially can be as large as the entire material, effectively converting the streaming application into a download-and-play application. In addition, the time needed to download the material can become so long that users lose interest while waiting for the playback to begin.

2) QoS Considerations: Guaranteed, high-quality delivery of real-time information over the Internet requires some form of resource reservations. Although there is ongoing work to bring such functionality to the Internet (via RSVP [64] and IPv6 as well as in MPEG-4), its global implementation is still a few years down the line. This is due to the fact that the network bandwidth is bound to stay as a limited resource needed by many in the foreseeable future, and therefore a value structure must be imposed on data exchanges with any defined QoS. Establishing such a structure may be easy for a corporate intranet; however, a nationwide implementation requires new laws, and a global implementation requires international agreements. An alternative to this, which can be implemented immediately, is to use the existing POTS telephone network, in particular ISDN, together with the Internet to provide a high QoS connection when needed.

Network on demand accomplishes a seamless integration of the telephone network and the Internet in order to provide an improved infrastructure for multimedia streaming applications. This is achieved by using the telephone network to bypass the congested parts of the Internet to provide a "private" network on demand with no changes to the user's applications. It can be considered as a special case of the FusionNet concept that covers several applications based on the joint use of connectionless networks with connection-oriented ones, e.g., ISDN and ATM. The best way to explain the network-on-demand concept is through a detailed description of a prototype implementation.

3) Network-on-Demand Prototype of FusionNet: Fig. 33 shows a prototype network-on-demand FusionNet system. A client is connected to an ISP through an ISDN connection, over which the PPP is employed for data communications. The implementation can also work with analog modems without any modifications. The ISP is assumed to have Internet access, modem pools for other customers, plus an internal network to connect its customers to each other and to the Internet. This part of the system is no different than the architectures being used to provide common Internet access today. The clients can access all Internet resources over this arrangement as usual. The difference is that a FusionNet-capable ISP can connect its clients to a FusionNet server using the telephone network when such a connection is needed.

In this particular implementation, the multimedia selections available on FusionNet servers are displayed on HTML pages made available to everyone on the Internet through hypertext transport protocol (HTTP) servers. When a client accesses such a page and wants to view a multimedia selection, he will be given the option to do this over the regular Internet connection or through a network-on-demand connection. For regular Internet delivery, the HTTP server merely conveys the FusionNet server's IP address to the client, which, in turn, uses the appropriate software to extract the multimedia material. In this case, the delivery path includes the FusionNet server's permanent Internet connection, which could be implemented using any one of the available Internet access techniques, the Internet segment between the server and the client's ISP, the ISP's internal network, and, last, the client's ISDN connection. As mentioned before, the performance over this path depends on the congestion on the Internet segments involved in this transmission and may or may not be acceptable.

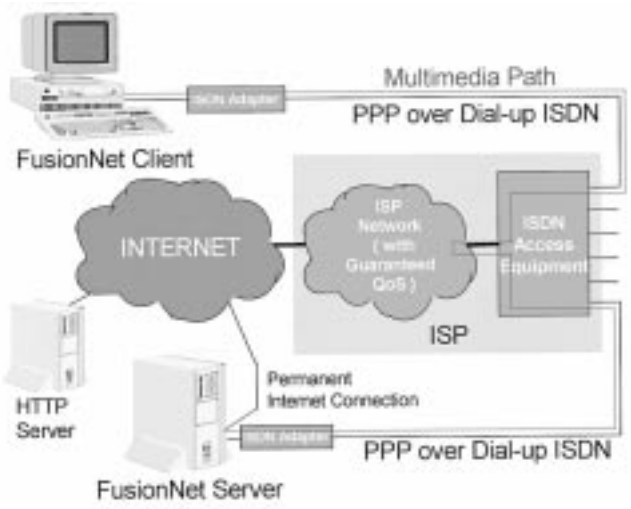

Fig. 33. Implementation of a prototype network-on-demand system.

If the network-on-demand option is selected, the HTTP server informs a FusionNet server via the Internet so that it establishes a dial-up, in this case an ISDN, connection to the client's ISP. Currently, the client's ISP is determined by posting an explicit form-based query on the client's browser when the client requests a network-on-demand-based delivery; however, an automatic determination based on the client's IP address may also be possible. After this connection gets established, the FusionNet server sends the IP number assigned to this new connection by the ISP to the HTTP server that initiated this transaction. The HTTP server uses this IP number for two purposes. First, it imbeds the number in all links on the HTML pages related to the multimedia material available to the client from this FusionNet server. This is needed to make sure that, when requested, the multimedia material gets delivered over this new network-on-demand connection instead of the FusionNet server's permanent Internet connection. Second, the HTTP server sends the IP number to the client, which, in turn, contacts the "connection manager" server running on the FusionNet server over the new connection. The Connection Manager keeps track of the connect time, which may be needed for billing purposes. Until a disconnect request, all multimedia transmission from the FusionNet server to the client is done through this newly established, mostly private, multimedia path, which contains no Internet segments.

The prototype is based on a software-only implementation at the application layer. On the client side, assuming the availability of the networking software and a WWW browser, the only additional software component needed is an application that gets launched when the client requests a network-on-demand connection. This application, called the "FusionNet agent," is responsible for receiving the FusionNet server's IP number from the HTTP server and contacting the FusionNet server's connection manager to confirm the connection establishment using TCP/IP sockets. It sends regular keep-alive messages to the connection manager during the session, and it sends a disconnect message at the end of the session. This feature was implemented as a client-side helper application that needs to be downloaded and configured as usual. If needed, it can easily be implemented as a "plug-in" or a "Java applet" so that the download and configuration steps can be skipped. After the connection gets established, retrieval and control of the multimedia material is done using any one of the several client-server applications available from various sources without any modifications.

The functionalities needed on the HTTP server side to support network on demand are implemented using a common gateway interface (CGI) script. This script implements the process responsible for displaying a form-based query on the client's browser when a network on demand is requested. It transmits this information to a FusionNet server and receives the IP number assigned to the server after the connection gets established. It sends this IP number to the client as an HTTP response defined as a special application. This launches the FusionNet agent on the client side. All communications are implemented using TCP/IP sockets. Also, the CGI script processes all HTML pages that have links to multimedia material contained on the FusionNet server before sending them to the client in order to imbed the server's IP number into these links. On the HTTP server, an additional server, called the FusionNet registry, is used to keep a current list of the available FusionNet servers. This server application listens at a known port for FusionNet server registry messages.

The networking software on a FusionNet server must be able to handle dual homing and dynamic address assignments. A PC running Microsoft's NT 4.0 operating system was used for this purpose. There are two main applications needed to implement network on demand on the FusionNet server side. The first application, called FusionNet main server, is responsible for registering the server to the FusionNet registries on selected HTTP servers when the FusionNet server becomes available. After the registration, the FusionNet main server listens to connection requests from these servers. Upon receiving a connection request, it dials into the specified ISP and, after the connection gets established, sends the assigned IP number back to the HTTP server. User authentication and determination of the necessary information for the ISP connection, such as the phone numbers, the user name, and the password, are carried out by the FusionNet Main Server also. The second application is the connection manager that registers the start of a session, listens to the keep-alive packets during the session, and handles disconnect messages coming from the client application. The connection manager is also responsible for dissolving the on-demand network when the keep-alive packets are missing. Also, the billing information for each session is saved by the connection manager.

Experimental versions of FusionNet have been running in the research laboratory for the past several years, based on ISDN connections to an ISP, and have provided good service for a wide range of audio/video material that has been found on the Web.

C. The CYBRARY Digital-Library Project

It may seem paradoxical to classify ordinary printed documents as multimedia, but equally paradoxical is the absence of a universal standard for efficient storage, retrieval, and transmission of high-quality document images in color. One of the promises (and anticipations) of digital libraries is to make the user experience "better than being there" in a real library. The SGML/HTML standard, and Adobe's PostScript and PDF formats, are appropriate for browsing through documents specifically created for electronic publishing, but they are notoriously inappropriate for representing images of previously existing and scanned documents. While the accuracy of OCR systems has steadily improved over the last decade, their performance is still far from being adequate to faithfully translate a scanned document into computer-readable format without extensive manual correction. Even if OCR accuracy were perfect, the visual aspect of the original document would be lost in the translated document. By way of example, consider the visual experience of seeing the Magna Carta in its original form and the "equivalent" visual experience of seeing the text of the Magna Carta in ordinary ASCII form. Visual details, including font irregularities, paper color, and texture are particularly important for ancient or historical documents, but the visual content and format are also crucial in documents with tables, illustrations, mathematical or chemical formulas, and handwritten text.

An obvious approach to creating virtually realistic documents is to store and transmit the documents as full-color images and to use OCR solely for the purpose of indexing on the basis of the machine-printed text within the document. When the document is remotely accessed, it is transmitted as a full-color image. We are now faced with two technical challenges. First, we need to devise a method for compressing document images that makes it possible to transfer a high-quality page over low-speed links (modem or ISDN) in a few seconds. Second, we need to devise an OCR system that is specifically geared toward the creation of an index.

The goal of the CYBRARY project at AT&T is to provide a virtual presence in a library by allowing any screen connected to the Internet to access and display high-quality images of documents. A second goal is to allow fast transmission of document images over low-speed connections while faithfully reproducing the visual aspect of the document, including color, fonts, pictures, and paper texture.

Several authors have proposed image-based approaches to digital libraries. The most notable example is the RightPages system [65]. The RightPages system was designed for document image transmission over a LAN. It was used for many years by the AT&T Bell Labs technical community. The absence of a universal and open platform for networking and browsing, such as today's Internet, limited the dissemination of the RightPages system. Similar proposals have been made more recently [66].

1) Displaying Documents: All of the above systems, and most commercially available document image-management systems, are restricted to black-and-white (bilevel) images. This is adequate for technical and business documents but insufficient for other types of documents such as magazines, catalogs, historical documents, or ancient documents. Many formats exist for coding bilevel document images, as discussed in the fax coding section (Section III-D2) of this paper, notably the G3 and G4 standards, the recent JBIG-1 standard, and the emerging JBIG-2 standard. Using the JBIG-2 standard, images of a typical black-and-white page at 300 dpi can be transmitted over a modem link in a few seconds. Work on bilevel image-compression standards is motivated by the fact that, until recently, document images were primarily destined to be printed on paper. Most low-cost printer technologies excel at printing bilevel images but have to rely on dithering and half-toning to print gray-level or color images, which reduces their effective resolution. However, the low cost and availability of high-resolution color displays is causing more and more users to rely on their screen rather than on their printer to display document images. Even modern low-end PC's can display 1024×768-pixel images with 16 b/pixel [5 b per red-green-blue (RGB) component], while high-end PC's and workstations can display 1280×1024 at 24 b/pixel. In the following discussion, document resolution numbers, such as "300 dpi," mean 300 dots per inch measured on the original paper document. The actual size of the document on the screen may vary with the screen size and resolution.

Most documents displayed in bilevel mode are readable at 200 dpi but are not pleasant to read. At 300 dpi, the quality is quite acceptable in bilevel mode. Displaying an entire 8.5×11 letter-size page at such high resolution requires a screen resolution of 3300 pixels vertically and 2500 pixels horizontally, which is beyond traditional display technology. Fortunately, using color or gray levels when displaying document images at lower resolutions dramatically improves the legibility and subjective quality. Most documents are perfectly readable, even pleasing to the eye, when displayed at 100 dpi on a color or grayscale display. Only documents with particularly small fonts require 150 dpi for effortless legibility. At 100 dpi, a typical page occupies 1100 pixels vertically and 850 pixels horizontally. This is within the range of today's high-end displays. Low-end PC displays have enough pixels, but in the wrong form factor.

2) Compression of Color Document Images: As stated earlier, the digital-library experience cannot be complete without a way of transmitting and displaying document images in color. In this section. we describe the "DJVU" format (pronounced "Déjà vu") for compressing high-quality document images in color. Traditional color image-compression standards such as JPEG (Section III-D3) are inappropriate for document images. JPEG's usage of local DCT's relies on the assumption that the high spatial frequency components in images can essentially be removed (or heavily quantized) without too much quality degradation. While this assumption holds for most pictures

Fig. 34. Document images with file sizes after compression in the DJVU format at 300 dpi.

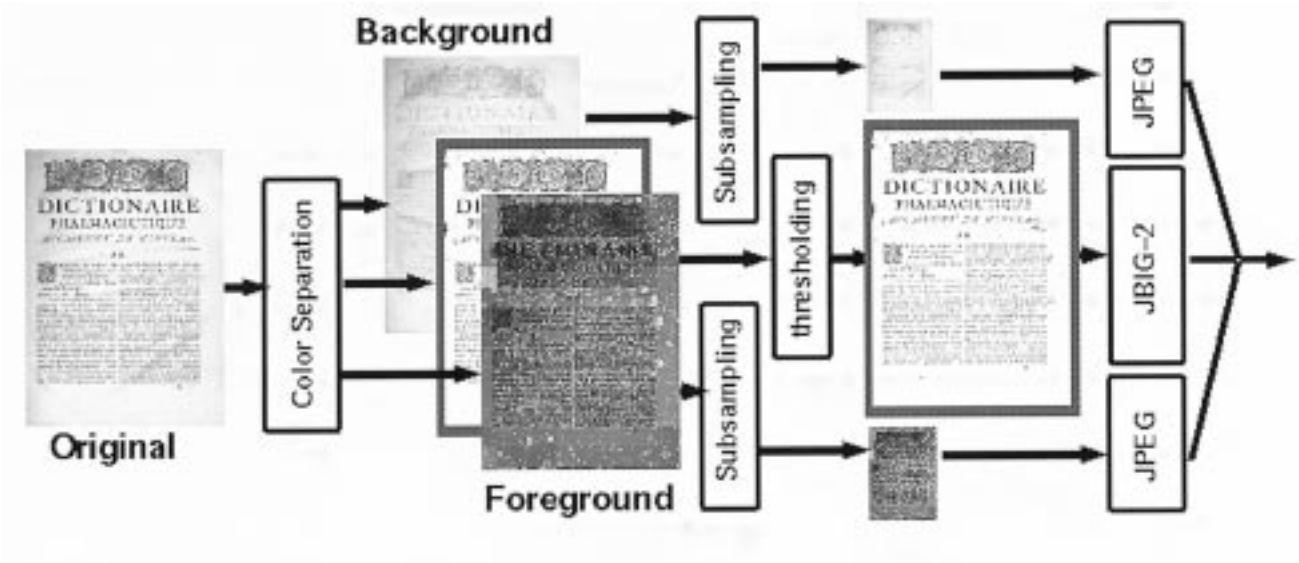

Fig. 35. The DJVU compression algorithm first runs the foreground/background separation. Both the foreground and background images are compressed using JPEG, while the binarized text and drawings are compressed using AT&T's proposal to the JBIG-2 standards committee.

of natural scenes, it does not hold for document images. The sharp edges of character images require a special coding technique so as to maximize readability.

It is clear that different elements in the color image of a typical page have different perceptual characteristics. First, the text is usually highly distinct from the background with sharp edges. The text must be rendered at high resolution, 300 dpi in bilevel or 100 dpi in color, if reading the page is to be a pleasant experience. The second type of element in a document image is the set of pictures. Rendering pictures at 50–100 dpi is typically sufficient for acceptable quality. The third element is the background color and paper texture. The background colors can be presented with resolutions less than 25 dpi for adequate quality.

Consider a document image, such as any of the ones shown in Fig. 34, scanned at 300 dpi with 24 bits/pixel. The main idea of the document image-compression technique is to generate and separately encode three images from which

the original image can be reconstructed: the background image, the foreground image, and the mask image. The first two are low-resolution (25 dpi) color images, and the latter is a high-resolution bilevel image (300 dpi). A pixel in the decoded image is constructed as follows.

- If the corresponding pixel in the mask image is zero, the output pixel takes the value of the corresponding pixel in the appropriately up-sampled background image.
- If the mask pixel is one, the pixel color is taken from the foreground image.

The foreground and background images can be encoded with any suitable means, such as JPEG. The mask image can be encoded using JBIG-2 (see Fig. 35).

3) Foreground/Background/Mask Separation: Consider the color histogram of a bicolor document image (i.e., an image with a quasi-uniform foreground and background color). Both the foreground and background colors are represented by peaks in this histogram. There may also be a small ridge between the peaks representing the intermediate colors of the pixels located near the character boundaries.

Extracting uniform foreground and background colors is easily achieved by running a clustering algorithm on the colors of all the pixels. The following algorithm, based on the well-known k-Means algorithm [67], accomplishes this task.

- Initialize the background color to white and the foreground color to black.
- Iterate over the pixels of the image and decide whether each is a foreground pixel or a background pixel by comparing the distances between the pixel color and the current foreground and background colors.
- Update the current foreground (or background) color by computing the average color of all the foreground (or background) pixels.
- Repeat steps 2) and 3) until convergence of the foreground and background colors. Although this convergence occurs very quickly, a careful use of stochastic approximations can make this algorithm even faster [68].

Naturally, there are more reliable initialization heuristics than simply assuming a white background and black foreground. For example, the highest peak in the smoothed color histogram can be assumed to be close to the background color.

4) Multiscale Block Bicolor Clustering: Typical document images are seldom limited to two colors. The document design and the lighting conditions induce changes in both the background and foreground colors over the image regions.

An obvious extension of the above algorithm consists of dividing the document image using a regular grid to delimit small rectangular blocks of pixels. Running the clustering algorithm within each block produces a pair of colors for each block. We can therefore build two low-resolution images whose pixels correspond to the cells of the grid. The pixels of the first image (or the second image) are painted with the foreground (or background) color of the corresponding block.

This block bicolor clustering algorithm is affected by several factors involving the choice of a block size and the selection of which peak of each block color histogram represents the background (or foreground) color.

Blocks should be small enough to capture the foreground color change, e.g., a red word in a line of otherwise black text. Therefore, the block should be, at most, as large as the largest character in the page.

Such a small block size, however, increases the number of blocks entirely located outside the text area. Such blocks contain only background pixels. Blocks may also be entirely located inside the ink of a big character. Such blocks contain only foreground pixels. In both cases, the clustering algorithm fails to determine a pair of well-contrasted foreground and background colors. A small block size also ruins the reliable heuristic algorithms for deciding which cluster center represents the foreground color and which one represents the background.

Instead of considering a single block size, we consider now several grids of increasing resolution. Each successive grid delimits blocks whose size is a fraction of the size of the blocks in the previous grid. By applying the bicolor clustering algorithm on the blocks of the first grid (the grid with the largest block size), we obtain a foreground and background color for each block in this grid. The blocks of the next-level grid are then processed with a slightly modified color clustering algorithm.

This modification biases the clustering algorithm toward choosing foreground and background colors for the small blocks that are close to the foreground and background colors found for the larger block at the same location. This bias is implemented using two methods: first, the colors obtained for the large blocks are used to initialize the clustering algorithm for the smaller block at the corresponding location; second, during the clustering algorithm, the cluster centers are attracted toward those initial colors.

This operation alleviates the small block size problem discussed above. If the current block contains only background pixels, for instance, the foreground and background colors of the larger block will play a significant role. The resulting background color will be the average color of the pixels of the block. The resulting foreground color will be the foreground color of the larger block. If, however, the current block contains pixels representing two nicely contrasted colors, the colors identified for the larger block will have a negligible impact on the resulting clusters.

5) Implementation: A fast version of this multiscale foreground/background color identification has been implemented and tested on a variety of high-resolution color images scanned at 24 b/pixel and 300 pixels/in.

The sequence of grids of decreasing block sizes is built by first constructing the highest resolution grid using a block size of 12×12 pixels. This block size generates foreground and background images at 25 pixels/in. Successive grids with decreasing resolutions are built by multiplying the

Table 7 ITU Multimedia Conferencing Standards

Standard	Network	Video	Audio	Multiplex	Control
H.320	ISDN	H.261	G.711*	H.221	H.242
H.323	LAN/Intranet	H.261	G.711*	H.225.0	H.245
H.324	POTS	H.263	G.723.1+	H.223	H.245

* G.722 and G.728 can also be used as the speech coders.

\+ G.723.1 can also be used as the speech coder.

block width and height by four until either the block width or height exceeds the page size.

This implementation runs in about 20 s on a 175-MHz MIPS R10000 CPU for a 3200 × 2200-pixel image representing a typical magazine page scanned at 300 pixels/in.

6) Color Document Image Compression with DJVU: Once the foreground and background colors have been identified with the above algorithm, a gray-level image with the same resolution as the original is computed as follows. The gray level assigned to a pixel is computed from the position of the projection of the original pixel color onto a line in RGB space that joins the corresponding foreground and background colors. If the projection is near the foreground, the pixel is assigned to black; if it is near the background, the pixel is white. This gray-level image will contain the text and all the areas with sharp local contrast, such as line art and drawings. This gray-level image is transformed into a bilevel image by an appropriate thresholding.

The document is now represented by three elements. The first two elements are the 25-dpi color images that represent the foreground and background color for each 12 × 12-pixel block. Those images contain a large number of neighboring pixels with almost identical colors. The third element is the 300-dpi bilevel image whose pixels indicate if the corresponding pixel in the document image should take the foreground or the background color. This image acts as a "switch" or stencil for the other two images.

To test the performance of this method, we have run experiments by compressing the 25-pixels/in foreground and background images using JPEG and compressing the 300 dpi bitmap using AT&T's proposal to the JBIG-2 standards committee, as described earlier.

We have selected ten images representing typical color documents, some of which are shown in Fig. 34. These images have been scanned at 300 pixels/in and 24 b/pixel from a variety of sources. Our compression scheme produces three files (two JPEG files for the foreground and background colors, one JBIG-2 file for the bitmap), whose combined size is reported in Table 7. The compression ratios range from 171 to 350. The readability of the reconstructed images is perfect at 300 pixels/in. On color screens, good readability is preserved even if the images are subsampled by a factor of three (with low-pass filtering) before being displayed.

7) Results and Comparison with Other Methods: Table 7 gives a full comparison of JPEG and the DJVU method on various documents. Compressing JPEG documents at 300 dpi with the lowest possible quality setting yields images that are of comparable quality with DJVU, but they are typically five to ten times bigger. For the sake of comparison, we subsampled the document images to 100 dpi (with local averaging) and applied JPEG compression with a 30% quality so as to produce files with similar sizes as with the DJVU format. It can be seen in Fig. 36 that JPEG presents many "ringing" artifacts that impede the readability.

We have also compared our method with Adobe's Acrobat/Capture product, which allows transcription of scanned images into the PDF format. Fig. 37 shows an example. The documents on Fig. 37 are standard OCR test documents provided by OCR Labs.[7] Acrobat/Capture performs the full OCR on the document and attempts to identify the font, point size, and character identity. It then produces a PDF file using fonts, point size, character positions, and identities that matches that of the original document as closely as possible. When the software assigns a low recognition score to an object, an original bitmap of the object is placed instead. The advantage of PDF is that the recognized characters can be printed at any resolution and will look clean. But there are several disadvantages. First, the mixing of original bitmaps and reconstructed fonts in a single page looks very unappealing. Second, because of undetected OCR errors, the PDF document may contain erroneous characters. Third, paper textures, texts of various colors, and nonuniform backgrounds are not handled. This makes it quite unsuitable for converting old documents, or for magazine pages with complex color schemes and layouts. The file sizes for a document encoded in DJVU are generally smaller than the same document encoded in PDF by Acrobat/Capture, although this depends very much on the original image quality. In another experiment, a black-and-white PostScript document generated by the LaTex/dvips word-processing system was rendered at 600 dpi, subsampled to 300 dpi, and run through the DJVU compressor. The resulting DJVU file was 48 Kbytes, while the original PostScript, after compression with "gzip," was 56 Kbytes.

8) Indexing Textual Image with OCR: As discussed in Section III-H, full-text-search techniques based on inverted word lists have been extensively studied in the information retrieval literature [52]. Building an index for a collection of textual images requires the user to perform document layout analysis and OCR. However, since in our system the OCR result is used solely for indexing, and has no effect on the appearance of the document on the user's screen, the requirements on the error rate are much less stringent than if the OCR were used to produce an ASCII transcription.

9) Simple Text Indexing: Simple, full-text search techniques on error-free text are commonly based on inverted word lists. A list of all the words that appear in the collection of documents is built. To each word is associated a list of documents in which the word appears. Various methods have been devised to efficiently encode those

[7] See http://onyx.com/tonymck/ocrlab.htm.

Fig. 36. Compression of (a) DJVU at 300 dpi and (b) JPEG at 100 dpi and quality factor 30%. The pictures are cut from [(a), left)] encyc2347.tif, [(b), left)] pharm1.tif, and [(a) and (b), right] hobby0002.tif. The file sizes are given in Table 7. The two methods give files of similar size but very different qualities.

lists so as to minimize their size while at the same time allowing easy Boolean set operations (union and intersection) and allowing updating when new documents are added to the collection. Common representations for the document lists include Golomb codes and methods for coding sparse bitmaps. In the bitmap representation, each word is associated with a bit string, where each bit indicates whether the word is present in a particular document. An explicit bitmap representation requires one bit per document, which is impractical for large collections. Efficient, compressed representations of sparse bitmaps have been devised that use trees represented as bit strings. Boolean set operations can easily be performed on such representations to execute complex queries.

Several authors have studied the effects of OCR errors on the accuracy of text-retrieval systems [69]. One of the main problems is that random errors tend to cause a significant increase in the size of the list of words, most of which are irrelevant. Lopresti and Zhou advocate the use of lists of subwords, which alleviates this problem but creates inefficiencies.

The next sections describes the document analysis and OCR system used in the CYBRARY project.

10) OCR for Index Creation: Before OCR can be performed, the document image must be analyzed and broken down into paragraphs, lines, and characters. First, a black-and-white image is created using the foreground/background/mask separation algorithm described above. Then, a connected component analysis is performed on this image. The result is a list of blobs (connected black pixels) associated with their coordinates and bounding box sizes. The following operations must be performed:

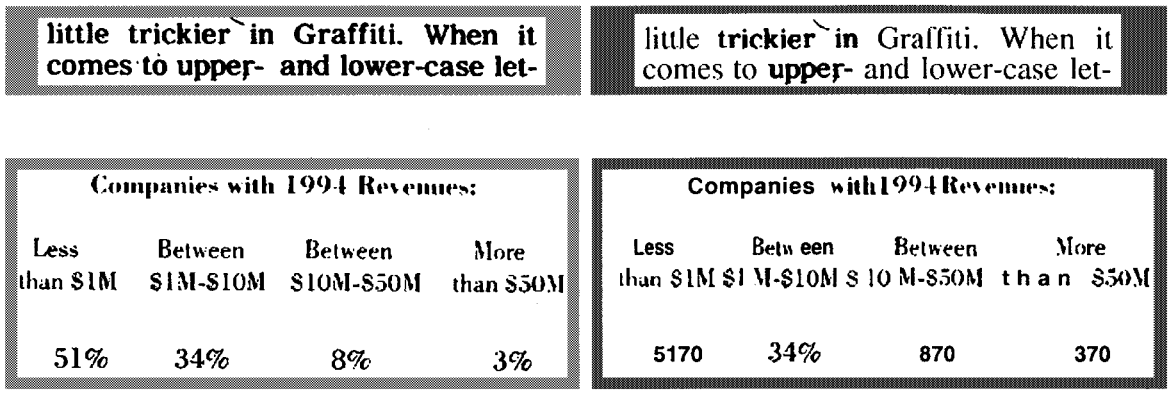

Fig. 37. Comparison of (a) DJVU-compressed documents with (b) Adobe PDF documents obtained with Acrobat/Capture.

a) Skew correction: The global orientation (or skew angle) of the image is measured with the following method. The list of coordinates of the bottom of each connected component is computed. Those points are then projected onto a vertical axis along various directions near the horizontal direction. The entropy of the projection histogram is then computed. The angle for which this entropy is minimum is searched using a simple iterative technique. The image is then deskewed using this angle.

b) Layout analysis: First, large components that are obviously not characters are eliminated from the image. Then, words and lines are found by iteratively grouping together connected components that are close to each other and approximately horizontally aligned. The procedure iteratively builds a tree data structure where the leaves contain single connected components and the nodes contain clusters of components grouped into words, lines, and paragraphs. The lines are then sent to the OCR system one by one.

c) Text line processing: To appropriately normalize the size of the characters before they are recognized, the baseline and the core height (the height of small lowercase characters such as "a") must be accurately measured. This is done by fitting a model to the text line. The model consists of four parallel lines whose relative positions and orientations are adjusted so that they fit local maxima and minima of the shape contours in the image as well as possible. The lower two lines will settle on the baseline points and the descender points, while the upper two lines will settle on the ascender points and top points of small characters. The fit is performed by an energy-minimization algorithm equivalent to the expectation-minimization algorithm applied to mixtures of Gaussians [70]. The line is then segmented into individual characters using heuristic image-processing techniques. Multiple segmentation hypotheses are generated and represented as a directed acyclic graph (DAG) (see Fig. 38).

d) Character recognition: The segmented characters are position and size normalized on a fixed-size image so that the baseline position is constant and the height of the character core (without ascenders and descenders) is constant. Those images are directly sent to the recognizer.

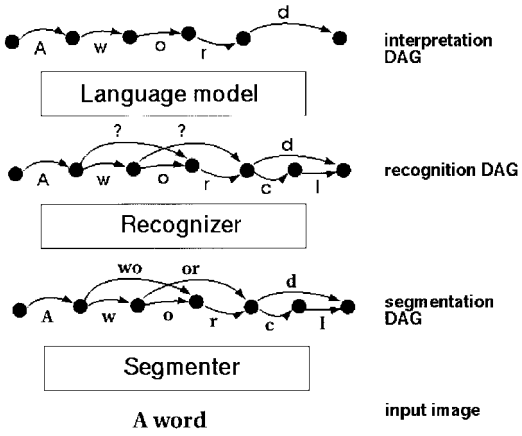

Fig. 38. Alternative segmentations of a line or word are represented by the paths of a DAG. Each segment is recognized, and the resulting DAG is combined with a grammar DAG to extract the best interpretation.

The recognizer is a convolutional neural network called LeNet5. Convolutional neural nets are specifically designed to recognize 2-D shapes with a high degree of invariance with respect to translations, scaling, skewing, and other distortions. They can directly accept images with no other preprocessing than approximate size normalization and centering. They have had numerous applications (some of which are commercial) in handwriting recognition [71]–[73] and object location in images, particularly faces [74]. The architecture of LeNet5 is shown in Fig. 39. In a convolutional net, each unit takes its input from a local "receptive field" in the previous layer, forcing it to extract a local feature. Units located at different places on the image are grouped in planes, called feature maps, within which units are constrained to share a single set of weights. This makes the operation performed by a feature map shift invariant and equivalent to a convolution followed by squashing functions. This weight-sharing technique greatly reduces the number of free parameters, thereby minimizing the necessary amount of training samples. A layer is composed of multiple feature maps that share different sets of weights, thereby extracting different features types.

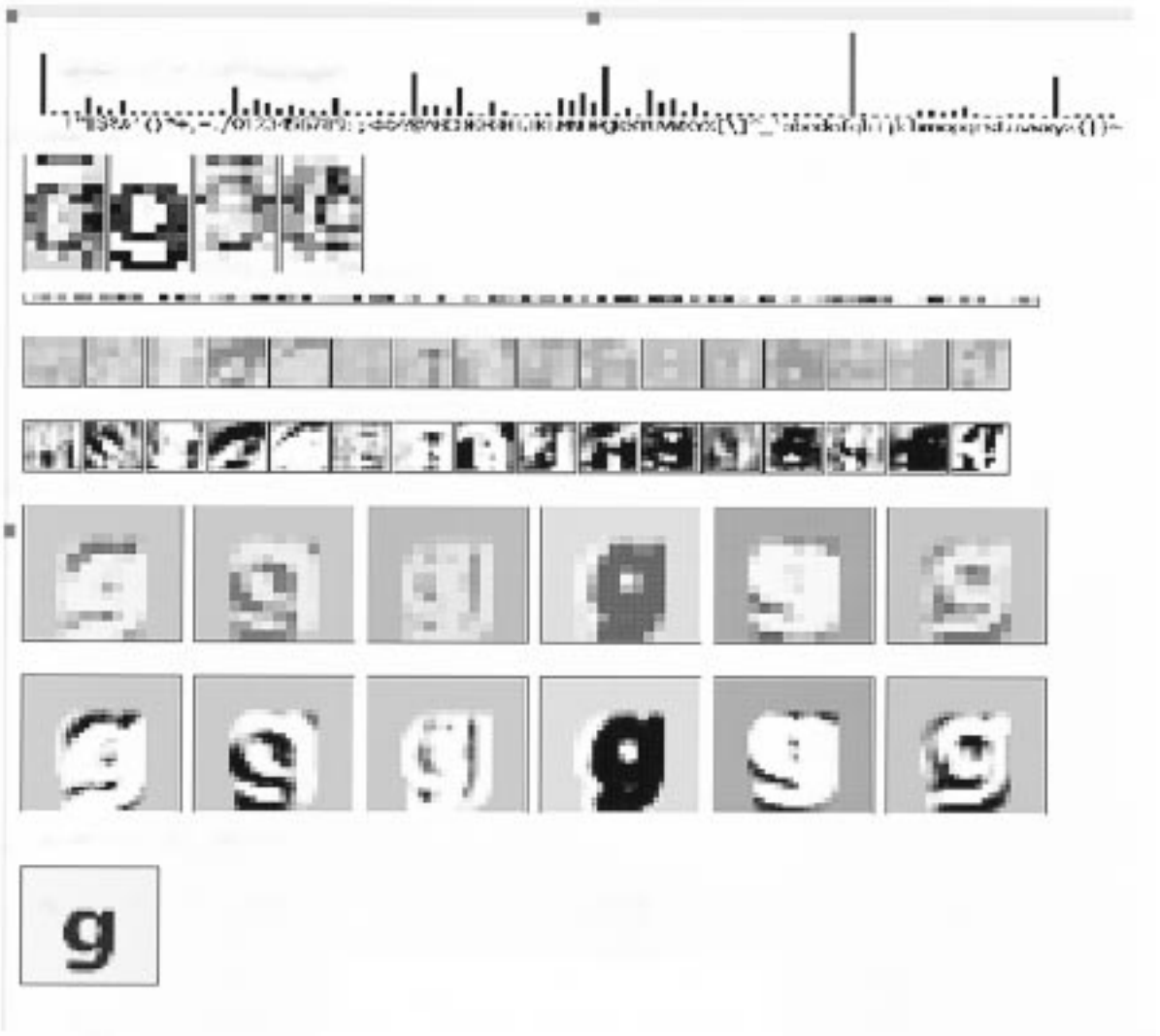

Fig. 39. The LeNet5 convolutional neural-network architecture for character recognition. LeNet5 comprises seven trainable layers.

The complete network is composed of multiple convolutional layers, extracting features of increasing complexity and abstraction. Reduced sensitivity to shifts and distortions is built into the system by inserting subsampling layers between the convolution layers. This forces the higher layers to extract progressively more global, and less position-sensitive, features. It is important to stress again that all the weights in such a network are trained by gradient descent; none of the extracted features are hand designed. The training process causes convolutional networks automatically to synthesize their own features. Computing the gradient is done with a slightly modified version of the classical back-propagation procedure. LeNet5 has 401 000 connections but only about 90 000 free parameters because of the weight sharing. It was trained with approximately 260 000 examples of characters of different fonts and point sizes covering the entire printable ASCII set. The error rates on fonts that were not used for training are 0.7% on uppercase letters, 1.3% on lowercase letters, 0.3% on digits, and 4.5% on ASCII symbols. The test set includes characters with very small point size scanned at 200 dpi. These numbers are therefore a pessimistic estimate of error rates on good-quality documents scanned at 300 dpi.

Each candidate segment is sent to the recognizer. The recognizer produces one label with an associated likelihood (or rather penalty that can be interpreted as a negative log-likelihood). If the character is ambiguous, the recognizer also produces alternative interpretations with their associated penalties. The range of possible interpretations for all the possible segmentations is again represented by a DAG. Each path in the DAG corresponds to one interpretation of one particular segmentation. The accumulated penalty for an interpretation is the sum of the penalties along the corresponding path. A shortest path algorithm, such as the Viterbi algorithm, can be used to extract the best interpretations (see Fig 38). The interpretation DAG can be combined with a language model (represented by a directed graph) to extract legal interpretations [75].

11) Indexing with Multiple Interpretations: The DAG representing the alternative interpretations of a word can be

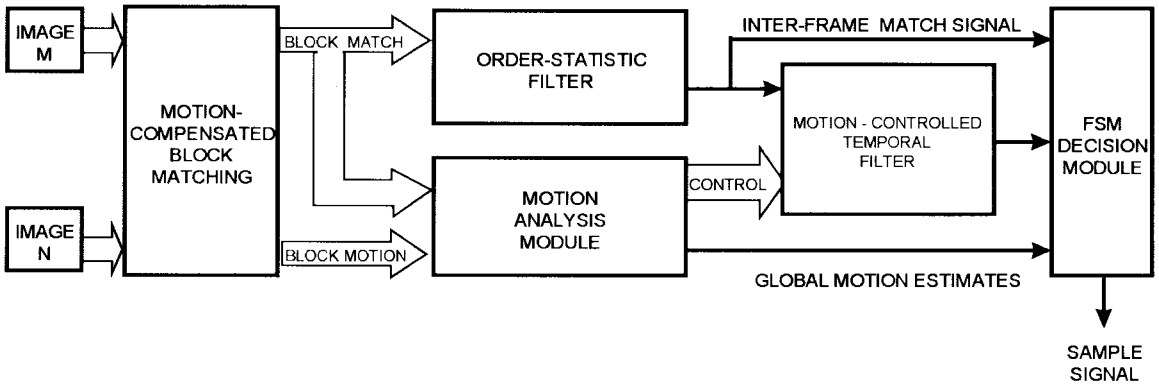

Fig. 40. Block diagram of content-based video sampling.

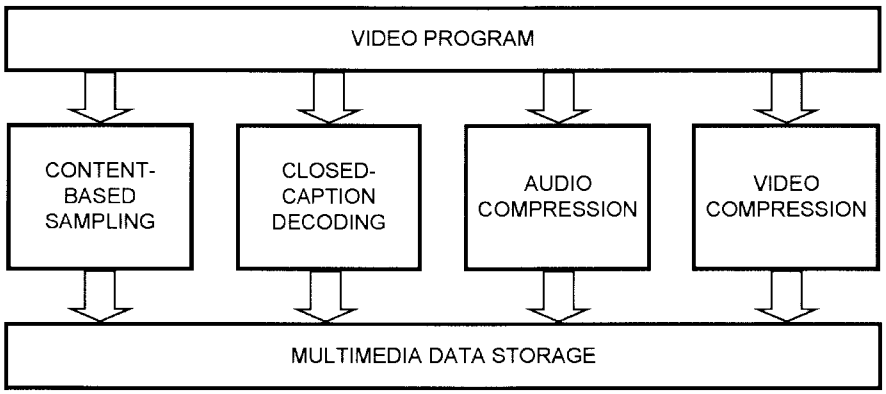

Fig. 41. Data-acquisition phase of the Pictorial Transcripts systems.

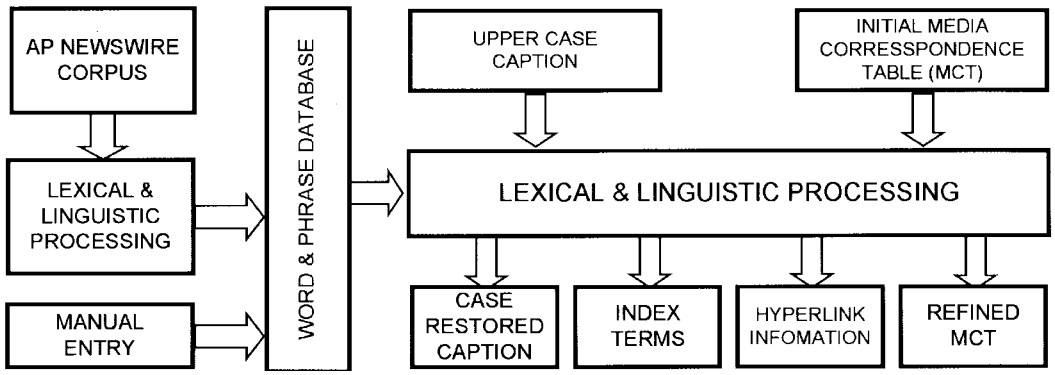

Fig. 42. Lexical and linguistic processing of closed-caption text.

used to improve the accuracy of a full-text search engine in the presence of OCR errors. When searching for a word in a document, the query word can be easily matched against all possible interpretations (represented by the DAG) for each word. For each match, the penalty of the corresponding path is returned. A generalization of this procedure can be used if the query word is itself represented by a DAG, i.e., when the query word is itself the result of a recognition. This can occur when the user clicks on a word in the document and asks for all matching words or when the query word is spoken as opposed to being typed or clicked upon. In that case, the query DAG and the word DAG are "composed" to produce a graph in which each path represents a word that is simultaneously present in the two input DAG's. Applying a shortest path algorithm to a composed graph extracts the best matching interpretation. This graph-composition operation has been well studied in the context of finite-state transducers used for language modeling [76].

D. Pictorial Transcripts

The Pictorial Transcripts system is an example of how some of the media-processing methods discussed in this paper can be applied to the fully automatic generation of a digital multimedia library of television programs [77]. This digital library currently contains more than 2500 hours of video programs in a condensed form that can be searched,

browsed, and selectively retrieved over communications networks.

Television and video programs are rich sources of information. Despite their dynamic nature, in comparison to documents consisting of text and still images, they lack the interactivity that is needed to perform efficient searching and browsing. Organizing these programs into a searchable and browsable digital library that is accessible on a communications network would provide a large archive of programs spanning a long period of time that would be available for selective retrieval and browsing.

A discussion of the media-processing algorithms and the searching and browsing capabilities of the Pictorial Transcripts system are given in the next sections.

1) Content-Based Sampling of Video: The bandwidth and storage requirements of digitized video (even in compressed form) present challenges to most multimedia server systems. This problem is addressed by representing the visual content of the program using a small set of still images that are selected by a content-based sampling algorithm [78]. The selection process divides the video into segments with similar visual contents and uses a single *representative frame* to represent each of the segments. In general, performing this task would require high levels of image understanding beyond what is possible, or feasible, with the current state of the art in computer vision and image understanding. However, the structure that has been built into professionally produced video programs can be exploited to achieve acceptable levels of performance. The content-based sampling method is based on detecting the abrupt and gradual transitions between consecutive shots, as well as quantitative measurement of camera motion parameters within individual shots.

The block diagram for the content-based video-sampling method is shown in Fig. 40. Motion-compensated block matching generates match and motion estimates between corresponding blocks in consecutive video frames. An order statistic filter combines the block-match values into a one-dimensional interframe match signal representing the likelihood that the consecutive frames belong to the same video segments (shots). This signal is a good indicator of abrupt transitions (i.e., *cuts*) between consecutive video shots. The detection of gradual transitions (e.g., dissolves, fades, digital editing, special effects) between shots requires additional processing of the interframe match signal over several video frames. This task is performed by temporally filtering the signal with a filter whose coefficients are controlled by a motion-analysis module. This enables the detection of a gradual transition while considerably reducing the likelihood of false detection resulting from rapid motion. The motion-analysis module also serves the purpose of estimating the components of the global motion resulting from camera motion. This camera motion information is used to detect *camera-induced* scene changes. These are changes in the visual contents of the scene that result from limited or continuous camera pan and tilt motions. Such motions are used in professionally produced video programs to achieve such goals as *association* (i.e., bridging

Fig. 43. A sample page from a Pictorial Transcript.

Fig. 44. Supplementary information about a topic.

the views from two different objects) or *orientation* (i.e., showing a wide area that would not fit in a single image). The camera-induced scene changes further divide individual shots into smaller segments from which representative frames are retained. A decision module employs a finite state machine to keep track of the sequence of events and generate a sample signal indicating the proper points within the video sequence where representative images should be selected. The decision module also serves the task of

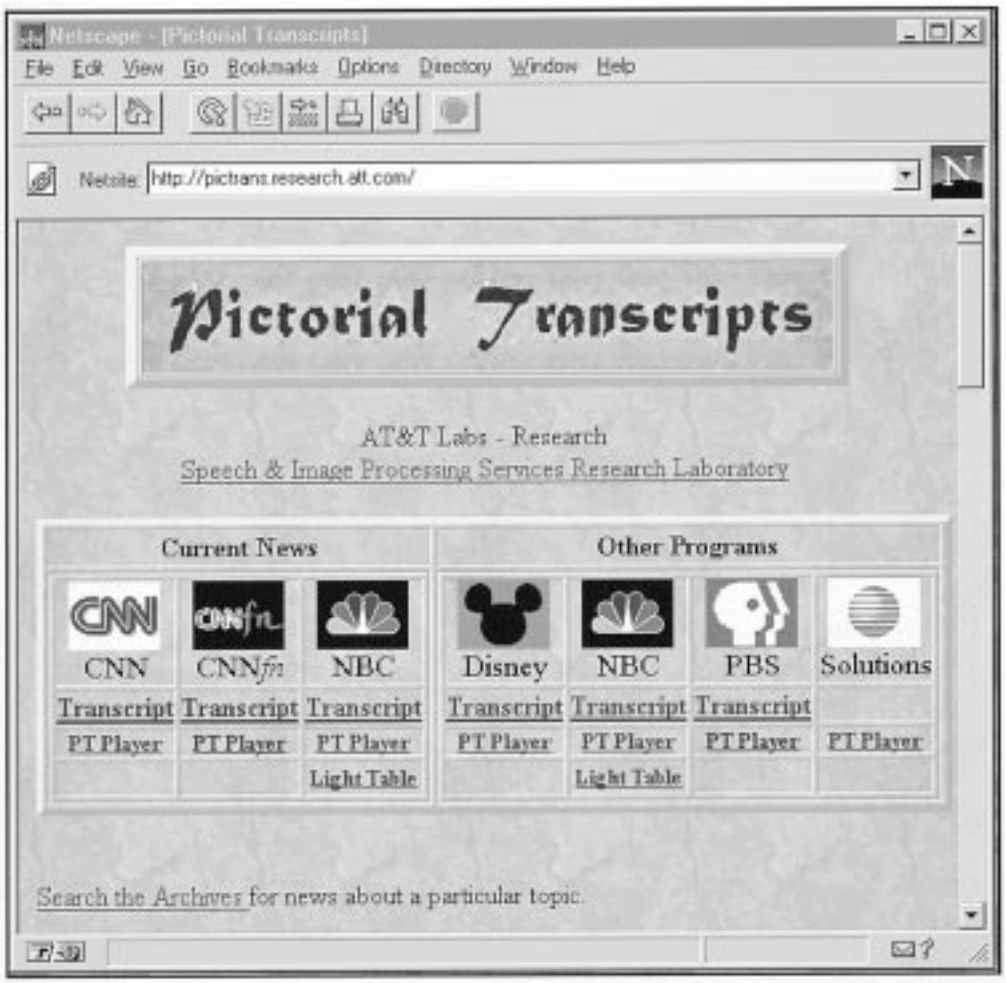

Fig. 45. The Pictoral Transcripts home page.

classifying the scene transition points into a number of categories (e.g., cut, fade, dissolve, camera induced, etc.) This information is utilized to improve the quality of the final presentation when choices need to be made among several representative images.

The set of images obtained by content-based video sampling provides a representation of the visual contents of a video program that can be used either in conjunction with other media, such as text or audio, to create a compact rendition of the program contents or as a pictorial index for selective retrieval of segments from the original program. Efficient computational methods have been devised that enable real-time implementation of the content-based sampling method using only a small fraction of the compute cycles on a PC. Additional image processing to compute the degree of similarity between different representative images, as well as spotting the occurrence of predetermined images in the video, are also performed. Such information is utilized to refine the final presentation and help create more intelligent searching and browsing mechanisms.

2) Text Processing: Textual representation of information enables the representation of nonpictorial information in a compact form that lends itself easily to text-based search. Ideally, such a *transcript* should be generated from the audio track of the program using ASR. LVASR's exist for specific domains (e.g., the North American Business News task), and they are capable of transcribing speech with an off-line word accuracy as high as 90.5% [79]. When applied to other tasks for which the existing language and acoustic models do not match well, however, the error rates can be considerably higher. Encouraging results have been obtained by applying such techniques to transcribing television programs [80]. Such transcripts have shown promise for searching and information-retrieval tasks. However, the high error rates limit the utility of such automatically generated transcripts for direct use by humans.

Manually entered closed-caption text that accompanies most television programs provides a more accurate textual representation of the information contained in the program. This information, which is included for the benefit of hearing-impaired viewers, is encoded in the video signal and is recovered from the video signal during the data-acquisition phase (see Fig. 41), decoded, and stored in raw form for further processing. This acquisition phase also includes real-time application of the content-based sampling method, as well as acquisition and compression of the audio and video components.

Fig. 46. The light table presentation of program content for visual searching.

(a) (b) (c)

Fig. 47. Navigation mechanisms in Pictorial Transcripts.

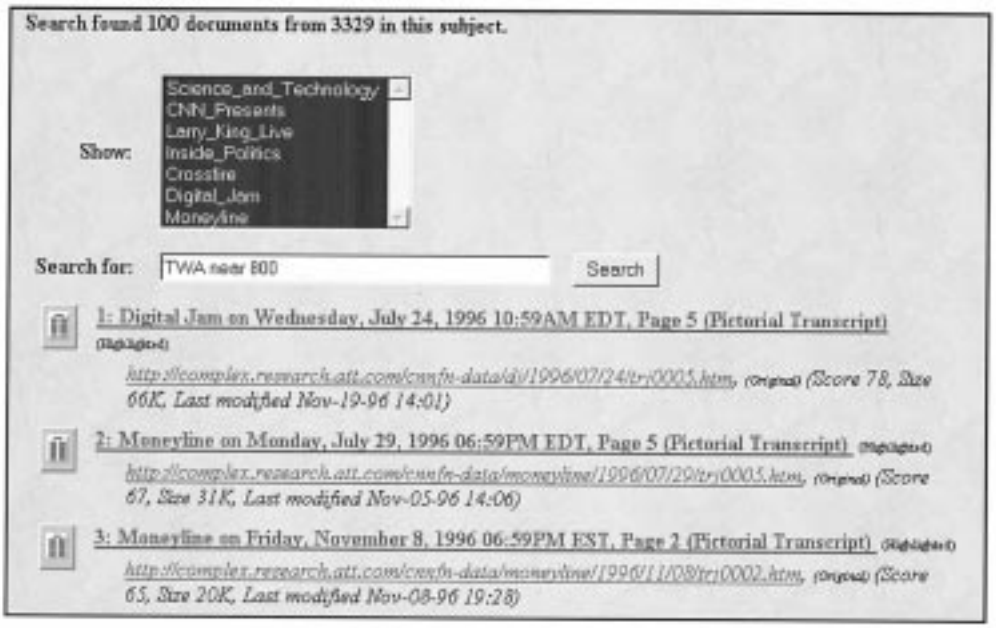

Fig. 48. An example of full-text search.

Fig. 49. An example of a pictorial index for a video program.

The raw closed-caption text undergoes lexical and linguistic processing (as shown in Fig. 42) to convert it to a form suitable for generating the hypermedia documents. This processing serves several purposes that are briefly outlined here. It converts all of the upper case text from the closed captions into lower case while preserving the correct capitalization. The processing also serves to extract key words and phrases for creating a list of index terms representing the content of the video. Another purpose of the text-processing stage is to extract relevant words or phrases that are used to create hyperlinks to other parts of the video or to other multimedia documents. As will be demonstrated later, such links are used to provide supplementary information about topics covered in the program. Currently, these tasks rely on phrase data bases that have been generated either automatically or manually prior to text processing. The manual entry is very limited and serves the purpose of supplying the

processing algorithms with specific information such as uncommon names and specific information about other available documents that can be used to provide supplementary information. The main phrase data base is generated automatically by an off-line analysis of a large corpus of text from the Associated Press newswire. Lexical processing also serves the goal of refining the temporal relationship (synchronization) between the images and the textual information. The synchronization information is recorded during the acquisition phase and stored in a media correspondence table.

3) Audio Encoding: While the combination of the content-sampled images and the associated text provides the most compact representation of the information content of the video program, audio is added to convey additional information about the video (e.g., music, nonspeech sounds). The audio component is essential when the closed-caption text is unavailable, and is helpful at all other times.

To provide a high-quality audio stream, a high-quality, wide-band speech coder [called the transform predictive coder (TPC)] is used to compress the 7-kHz-bandwidth audio stream to a 16-kb/s rate. This bit rate can be reliably transmitted (along with the images and the text) over switched telephone networks with commonly available modems running at rates above 28.8 kb/s. Digitization and compression of the audio signal is performed in real time during the acquisition phase. The audio signal is segmented on the basis of its temporal relationship with the video segments. In the case when the video program contains CD-quality audio, a somewhat higher rate audio coder, such as the PAC or the AAC algorithm described in Section III-C2, could be used to preserve the higher bandwidth, higher quality signal.

4) Automated Hypermedia Document Generation: In the final phase, the information derived by media processing is used to automatically generate a hypermedia rendition of the video program contents in HTML form. It should be stressed that the process of generating the Pictorial Transcripts takes place in real time and in a fully automatic manner. Each program is organized into several HTML pages to make it more manageable. Currently, the segmentation of the program into individual pages is based on size as well as detected commercial boundaries. Eventually, the system will be able to take advantage of more advanced text-segmentation techniques to perform topic segmentation. A segment of a sample HTML page generated by the system is shown in Fig. 43. The images extracted by the content-based sampling algorithm are displayed next to the corresponding linguistically processed closed-caption text.

Audio icons next to each image can be used to initiate (or terminate) the replay of the corresponding audio contents. Hence, the compact representation serves as an index into the audio stream associated with the video program. A similar arrangement is used to initiate selective replay of motion video by selecting the images. Operating in conjunction with a video delivery system (such as FusionNet, discussed in Section IV-B), the searching and browsing capabilities of the Pictorial Transcripts are used to select a video segment, which is then delivered with guaranteed QoS by the FusionNet server. Fig. 43 also depicts the use of the automated word and phrase spotting to generate links to supplementary information. The two (underlined) occurrences of the phrase "United States" on this page are links to another HTML document with additional information. This link takes the user to a document in a different digital library (i.e., the *CIA World Factbook*), a small segment of which is shown in Fig. 44.

5) Searching and Browsing Capabilities: The research prototype of the Pictorial Transcripts system currently contains more than 2500 hours of video programs from several broadcasters, as well as some internal video presentations. Effective utilization of this large digital multimedia library calls for intelligent mechanisms for selective retrieval of information. This is done by organizing the programs into several different categories and providing several searching and browsing mechanisms. Fig. 45 shows a segment of the Pictorial Transcripts home page.

This figure shows the organization of the programs based on the television network. There are several different representations for each program. The "transcript" gives an HTML presentation of the information in the form that consists of still images, text, and icons for initiating the replay of associated audio. This representation is depicted in Fig. 43. The "PT player" representation provides a real-time replay of the same image, text, and audio information using a streaming player that will be discussed later. The "light table" rendition of a segment of a video program is depicted in Fig. 46. This representation involves images, or images and audio, and is useful for rapid browsing of the program material using pictures only. This representation enables users quickly to browse through the entire program to spot an image of interest. The audio icon can then be used to supply additional information about the program by initiating the replay of audio at the corresponding point.

Selecting one of the networks (e.g., CNN) from the home page presents the user with another page indicating the different programs from the selected network [Fig. 47(a)]. For each program, the option of selecting the most recent version will take the user directly to that program. Selecting the "previous shows" will give the user access to the archives of the program through a sequential calendar interface depicted in Fig. 47(b) and (c).

The system provides several searching and browsing mechanisms to facilitate selective retrieval of specified material. One such searching mechanism allows the user to select programs based on full-text search, as shown in Fig. 48. This interface enables the user to specify one or a combination of programs to be included in the search. A search string specified by the user returns information about relevant video programs that can be selectively retrieved. As mentioned earlier, each video program is organized into pages. Each page provides links to enable navigation

Fig. 50. An example of the keyword section of an index page.

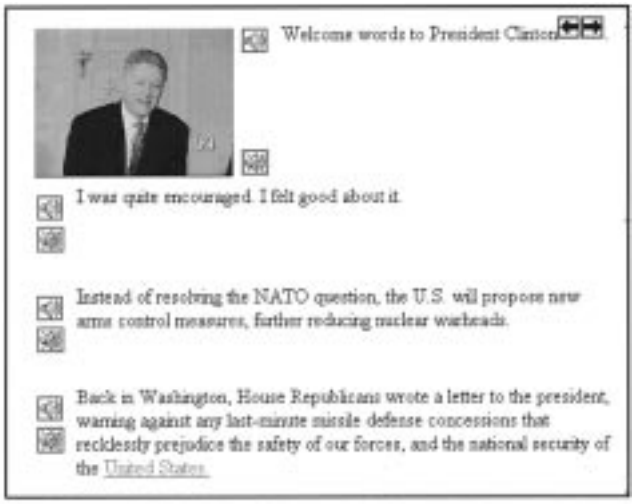

Fig. 51. A browsable keyword-specific transcript.

to the previous, next, or any other page in the program, as well as an *index page*. The index page enables the selection of the pages using textual and pictorial indexes. Segments of an index page are shown in Figs. 49 and 50.

The pictorial index (Fig. 49) consists of a collection of images, each of which represents the contents of a single page of the transcript. These images can be clicked to bring up the corresponding page of the transcript. The index page (Fig. 50) enables the user to retrieve a given segment of the program by specifying the time. It also presents the user with a set of key words and phrases extracted during the linguistic processing phase of the Pictorial Transcripts generation. Selecting one of these words or phrases results in the generation of a *keyword-specific transcript* that is geared toward the efficient browsing of the entire program to access relevant information about the selected keyword.

Added efficiency is achieved by displaying only a subset of the representative images that are likely to correspond to the selected keyword. The association is established based on proximity (i.e., only images that correspond with, or are in close proximity to, the segments of text containing the keyword are retained). The keyword-based browsing capability is achieved by placing navigation controls next to each occurrence of the keyword in the specialized transcript. These controls allow the user to jump to preceding or succeeding occurrences of the word in the transcript. A segment of a keyword-specific transcript generated by selecting the phrase "President Clinton" (in Fig. 50) is shown in Fig. 51. In this case, the proximity-based association criterion performs satisfactorily and presents the user with a relevant picture. The forward arrow takes the user to the next occurrence of the phrase, while the backward arrow goes to the previous occurrence.

6) Streaming Delivery: The representation of the information contents of video programs in standard HTML form enables the user to take advantage of standard HTML viewers to access the information. A representation using text and a small number of still images is perfectly suitable for representation in HTML form. The addition of a continuous medium such as audio to the representation, however, imposes new constraints on the system in terms of real-time delivery of audio. As discussed in detail in Section III-C4, such continuous media can be either downloaded in their entirety and then played or delivered in streaming mode. Fig. 52 shows a streaming player for the Pictorial Transcripts. This player delivers a real-time presentation of the video program using the content-based sampled images in synchronization with linguistically processed closed-captioned text and audio. The combination of JPEG compressed images, TPC encoded audio, and Lempel–Ziv compressed text results in sufficiently low bandwidth requirements so that the resulting multimedia stream can be delivered reliably over a dialed-up 28.8-kb/s modem connection. The player provides VCR-type controls for browsing. The forward and backward controls enable nonlinear navigation through the program by jumping to the next or previous scene. The slider allows rapid image-based browsing of the program.

V. Summary

In this paper, we have discussed a range of technological problems that define multimedia systems and multimedia processing as they have evolved over the past decade. The original concept of multimedia was any application or system involving signals from two or more media, including text, speech, audio, images, video, handwriting, and data. This narrow view of multimedia has evolved to include all aspects of multimedia processing, including comput-

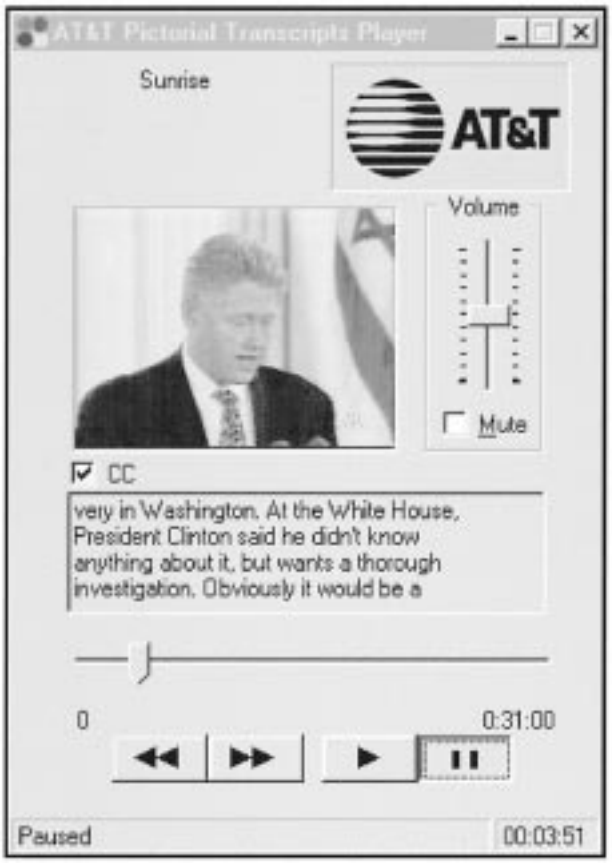

Fig. 52. The streaming multimedia player.

ing, communications, and networking for storage, delivery, transport, access, indexing, searching, and browsing. Since multimedia systems include both person-to-person (e.g., teleconferencing) and person-to-machine applications (e.g., searching and browsing stored multimedia content), key technological issues include specification and implementation of appropriately defined user interfaces to both multimedia systems and the networks that store and transport multimedia source material. Last, the emergence of data networks such as the Internet have redefined multimedia systems concepts such as data streaming, multiplexing (both real-time and nonreal-time streams), control, and conformance testing. This has led to the evolution of modern standards that specify methods for compression and coding of speech, audio, image, and video signals, multiplexing of data streams, systems design and evaluation, conformance testing, and system compatibility and control. These modern standards fit in well with modern data-transport protocols such as ATM, frame relay, TCP/IP, and UDP, all of which provide capability for a number of highly varying data types to coexist over a packetized network. As the modern telecommunications network evolves to include guaranteed QoS for synchronous data, such as speech, audio, and video (much as has been the case for voice on the POTS network for the past 100 years), powerful multimedia systems, of the types described in this paper, will evolve and become commonplace and widely used over the next decades.

ACKNOWLEDGMENT

The authors wish to acknowledge that the original work described in this paper was done by a number of individuals at AT&T Laboratories as well as outside of AT&T. They also wish to acknowledge both the strong contributions to the literature and the fine work done by the following individuals, as reported on in this paper: J. Johnston and S. Quackenbush (audio coding), P. Howard (fax and image coding), A. Puri, J. Ostermann, T. Chen, and A. Reibman (video coding), M. R. Civanlar, D. Gibbon, and J. Snyder (streaming of audio and video), C. Kamm and M. Walker (user interface design), M. R. Civanlar and G. Cash (FusionNet), L. Bottou, P. Simard, and P. Howard (CYBRARY), and D. Gibbon (Pictorial Transcripts). The authors wish to thank each of these individuals for their assistance in providing the source material on which this tutorial paper is based. They also thank the two informal reviewers, D. Roe and W. Rabiner, for their thorough reading and numerous valuable comments and corrections, as well as the formal IEEE reviewers.

REFERENCES

[1] E. B. Carne, *Telecommunications Primer: Signals, Building Blocks and Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1995.

[2] N. Kitawaki and K. Itoh, "Pure delay effects on speech quality in telecommunications," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 586–593, May 1991.

[3] R. R. Riesz and E. T. Klemmer, "Subjective evaluation of delay and echo suppressors in telephone communications," *Bell Syst. Tech. J.*, vol. 42, pp. 2919–2941, Nov. 1963.

[4] J. R. Cavanaugh, R. W. Hatch, and J. L. Sullivan, "Models for the subjective effects of loss, noise, and talker echo on telephone connections," *Bell Syst. Tech. J.*, vol. 55, pp. 1319–1371, Nov. 1976.

[5] U. Black, *ATM: Foundation for Broadband Networks.* Englewood Cliffs, NJ: Prentice-Hall, 1995.

[6] B. Irving, "Real-time Internet—Challenges and opportunities in making Internet telephony pervasive," presented at Euro Forum, The Netherlands, Dec. 5, 1996.

[7] "One way transmission time," International Telecommunications Union, Geneva, Switzerland, Recommendation ITU-T G.114, 1996.

[8] W. B. Kleijn and K. K. Paliwal, Ed., *Speech Coding and Synthesis.* Amsterdam, The Netherlands: Elsevier, 1995.

[9] N. O. Johannesson, "The ETSI computation model—A tool for transmission planning of telephone networks," *IEEE Commun. Mag.*, pp. 70–79, Jan. 1997.

[10] A. V. McCree, K. Truong, E. B. George, T. P. Barnwell, and V. Viswanathan, "A 2.4 kbit/s MELP coder candidate for the new U.S. federal standard," in *Proc. ICASSP'96*, May 1996, pp. 200–203.

[11] H. Fletcher, "Loudness, masking and their relationship to the hearing process and the problem of noise measurement," *J. Acoust. Soc. Amer.*, vol. 9, pp. 275–293, Apr. 1938.

[12] B. Scharf, "Critical bands," in *Foundations of Modern Auditory Theory*, J. Tobias, Ed. New York: Academic, 1970, pp. 159–202.

[13] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models.* New York: Springer-Verlag, 1990.

[14] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Select. Areas Commun.*, vol. 6, pp. 314–323, Feb. 1988.

[15] K. Brandenburg and M. Bosi, "Overview of MPEG-audio: Current and future standards for low-bit rate audio coding," *J. Audio Eng. Soc.*, vol. 45, nos. 1/2, pp. 4–21, Jan./Feb. 1997.

[16] R. W. Baldwin and C. V. Chang, "Locking the e-safe," *Spectrum*, pp. 40–46, Feb 1997.

[17] M. A. Sirbli, "Credits and debits on the Internet," *Spectrum*, pp. 23–29, Feb. 1997.

[18] M. R. Civanlar, submitted for publication.

[19] A. N. Netravali and B. G. Haskell, *Digital Pictures—Representation, Compression, and Standards*, 2nd ed. New York: Plenum, 1995.

[20] K. R. McConnell, D. Bodson, and R. Schaphorst, *FAX, Digital Facsimile Technology and Applications.* Boston, MA: Artech House, 1992.

[21] W. B. Pennebaker and J. L. Mitchell, "Other image compression standards," in *JPEG Still Image Data Compression Standard*. New York: Van Nostrand, 1993, ch. 20.

[22] *Progressive Bi-Level Image Compression*, ISO/IEC International Standard 11544, 1993.

[23] K. Mohiudin, J. J. Rissanan, and R. Arps, "Lossless binary image compression based on pattern matching," in *Proc. Int. Conf. Computers, Systems, and Signal Processing*, Bangalore, India, 1984, pp. 447–451.

[24] P. G. Howard, "Lossy and lossless compression of text images by soft pattern matching," in *Proceedings of the Data Compression Conference*, J. A. Storer and M. Cohn, Ed. Snowbird, UT: IEEE Press, 1996, pp. 210–219.

[25] R. N. Ascher and G. Nagy, "A means for achieving a high degree of compaction on scan-digitized printed text," *IEEE Trans. Comput.*, vol. C-23, pp. 1174–1179, Nov. 1974.

[26] P. G. Howard, "Text image compression using soft pattern matching," *Computer J.*, 1997.

[27] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electron.*, vol. 38, p. 34, Feb. 1992.

[28] *Digital Compression and Coding of Continuous-Tone Still Images*, ISO/IEC International Standard 10918-1, 1991.

[29] W. B. Pennebaker and J. L. Mitchell, "JPEG coding models," in *JPEG Still Image Data Compression Standard.* New York: Van Nostrand, 1993, ch. 10.

[30] "Call for contributions for JPEG-2000," ISO/IEC JTC1/SC29/WG1 N505, 1997.

[31] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circ. Syst. Video Technol.*, vol. 6, pp. 243–250, 1996.

[32] J. L. Mitchell, W. B. Pennebaker, C. E. Fogg, and D. J. LeGall, *MPEG Video Compression Standard.* New York: Chapman and Hall, 1997.

[33] H.-M. Hang and J. W. Woods, *Handbook of Visual Communications.* New York: Academic, 1995.

[34] "Video codec for audiovisual services at p*64 kbits/sec," International Telecommunications Union, Geneva, Switzerland, Recommendation H.261, 1990.

[35] "Video coding for low bit rate communication," International Telecommunications Union, Geneva, Switzerland, Recommendation H.263, 1995.

[36] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2.* New York: Chapman and Hall, 1997.

[37] A. Puri, R. Aravind, and B. G. Haskell, "Adaptive frame/field motion compensated video coding," *Signal Process. Image Commun.*, vol. 1–5, pp. 39–58, Feb. 1993.

[38] C. Huitema, *IPv6: The New Internet Protocol.* Englewood Cliffs, NJ: Prentice-Hall, 1996.

[39] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human–Computer Interaction.* Menlo Park, CA: Addison-Wesley, 1986.

[40] L. R. Rabiner, "Applications of voice processing to telecommunications," *Proc. IEEE*, vol. 82, pp. 199–228, Feb. 1994.

[41] R. Sproat and J. Olive, "An approach to text-to-speech synthesis," in *Speech Coding and Synthesis*, W. B. Kleijn and K. K. Paliwal, Eds. Amsterdam, The Netherlands: Elsevier, 1995, pp. 611–633.

[42] J. P. Van Santen, R. W. Sproat, J. P. Olive, and J. Hirschberg, Eds., *Progress in Speech Synthesis.* Berlin, Germany: Springer-Verlag, 1996.

[43] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[44] L. R. Rabiner, B. H. Juang, and C. H. Lee, "An overview of automatic speech recognition," in *Automatic Speech and Speaker Recognition, Advanced Topics*, C. H. Lee, F. K. Soong, and K. K. Paliwal, Eds. Norwell, MA: Kluwer, 1996, pp. 1–30.

[45] A. Gorin, G. Riccardi, and J. Wright, "How may I help you?" *Speech Commun.*, to be published.

[46] S. Boyce and A. L. Gorin, "User interface issues for natural spoken dialogue systems," in *Proc. Int. Symp. on Spoken Dialogue (ISSD)*, 1996, pp. 65–68.

[47] E. Levin and R. Pieraccini, "CHRONUS, the next generation," in *Proc. 1995 ARPA Spoken Language Systems Technology Workshop*, Austin, TX, pp. 269–271, 1995.

[48] S. Bennacef, L. Devillers, S. Rosset, and L. Lamel, "Dialog in the RAILTEL telephone-based system," in *Proc. ISSD'96*, pp. 173–176.

[49] A. Abella, M. K. Brown, and B. Buntschuh, "Development principles for dialog-based interfaces," in *Proc. ECAI-96 Spoken Dialog Processing Workshop*, Budapest, Hungary, 1996, pp. 1–7.

[50] C. A. Kamm, S. Narayanan, D. Dutton, and R. Ritenour, submitted for publication

[51] R. M. Krauss and P. D. Bricker, "Effects of transmission delay and access delay on the efficiency of verbal communication," *J. Acoust. Soc. Amer.*, vol. 41.2, pp. 286–292, 1967.

[52] J. Cowie and W. Lehnert, "Information extraction," *Commun. ACM*, vol. 39, no. 1, pp. 80–91, Jan. 1996.

[53] G. Salton, A. Singhal, C. Buckley, and M. Mitra, "Automatic text decomposition using text segments and text themes," in *Proc. 7th ACM Conf. Hypertext*, Washington, DC, Mar. 1996, pp. 53–65.

[54] M. Mohri and M. Riley, "Weighted determinization and minimization for large vocabulary speech recognition," in *Proc. Eurospeech'97*, pp. 131–134.

[55] M. Christel, T. Kanade, M. Mouldin, R. Reddy, M. Sirbu, S. Stevens, and H. Wackler, "Informedia digital video library," *Commun. ACM*, vol. 38, no. 4, pp. 57–58, Apr. 1995.

[56] L. R. Rabiner, "A tutorial on hidden Markov models and its application to speech recognition," in *Proc. IEEE*, vol. 77, pp. 257–286, Feb. 1989.

[57] J. G. Wilpon, L. R. Rabiner, C. H. Lee, and E. Goldman, "Automatic recognition of keywords in unconstrained speech using hidden Markov models," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 1870–1878, Nov. 1990.

[58] M. G. Brown, J. T. Foote, G. J. Jones, K. S. Jones, and S. A. Young, "Open vocabulary speech indexing for voice and video mail retrieval," in *Proc. 4th Int. Conf. Multimedia*, Nov. 1996, pp. 307–316.

[59] D. Roy and C. Malamud, "Speaker identification based on text-to-audio alignment for an audio retrieval system," in *Proc. ICASSP'97*, Munich, Germany, May 1997, pp. 1099–1102.

[60] I. Witten, A. Moffat, and T. Bell, *Managing Gigabytes, Compressing and Indexing Documents and Images.* New York:

Chapman and Hall, 1994.

[61] D. Salesin, "Image editing, image querying, and video clip art: Three multi-resolution applications in computer graphics," in *Invited Plenary Talk at ICASSP'96*, Atlanta, GA, May 1996, pp. 431–433.

[62] M. R. Civanlar, G. L. Cash, and B. G. Haskell, "FusionNet: Joining the Internet and phone networks for multimedia applications," in *Proc. ACM Multimedia*, Boston, MA, Nov. 1996, pp. 431–433.

[63] H. Schulzrinne, A. Rao, and R. Lanphier, "Real-time streaming protocol (RTSP)," Internet Engineering Task Force, Internet draft, draft-ietf-mmusic-rtsp-01.txt, Feb. 1997.

[64] L. Zhang, S. Berson, S. Herzog, and S. Jamin, *Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification*, Internet Engineering Task Force, Internet draft, draft-ietf-rsvp-spec-14.txt, Nov. 1996.

[65] G. Story, L. O'Gorman, D. Fox, L. Shaper, and H. Jagadish, "The right pages image based electronic library for alerting and browsing," *IEEE Comput. Mag.*, vol. 25, no. 9, pp. 17–26, 1992.

[66] T. Phelps and R. Wilensky, "Toward active, extensible, networked documents: Multivalent architecture and applications," in *Proc. 1st ACM Int. Conf. Digital Libraries*, Bethesda, MA, 1996, pp. 100–108.

[67] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the Fifth Berkeley Symp. on Mathematics, Statistics, and Probabilities, 1.*, L. LeCam and J. Neyman, Eds. Berkeley, CA: Univ. of California Press, 1967, pp. 281–297.

[68] L. Bottou and Y. Bengio, "Convergence properties of the K-means algorithm," *Advances in Neural Information Processing Systems 7.* Cambridge, MA: MIT Press, 1995.

[69] D. Lopresti and L. Zhou, "Retrieval strategies for noisy text," in *Proc. Advances Digital Libraries*, 1996, pp. 76–85.

[70] Y. Bengio and Y. LeCun, "Word normalization for on-line handwritten word recognition," in *Proc. Int. Conf. Pattern Recognition*, Jerusalem, Israel, Oct. 1994, pp. 409–413.

[71] Y. LeCun, O. Matan, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten zip code recognition with multilayer networks," in *Proc. Int. Conf. Pattern Recognition*, Atlantic City, NJ, Apr. 1990, pp. 35–40.

[72] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, M. Arbib, Ed. Cambridge, MA: MIT Press, 1995.

[73] Y. Bengio, Y. LeCun, C. Nohl, and C. Burges, "LeRec: A NN/HMM hybrid for on-line handwriting recognition," *Neural Comput.*, vol. 7, no. 6, pp. 1289–1303, Nov. 1995.

[74] R. Vaillant, C. Monrocq, and Y. LeCun, "An original approach for the localization of objects in images," in *Proc. IEEE Vision, Image, and Signal Processing*, vol. 141, no. 4, Aug. 1994, pp. 245–250.

[75] Y. LeCun, L. Bottou, and Y. Bengio, "Reading checks with graph transformer networks," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Munich, Germany, 1997, pp. 151–154.

[76] M. Mohri, F. Pereira, and M. Riley, *Syntactic Algorithms, Language, Speech and Sequence Processing by Automata and Transducers.* Cambridge, MA: MIT Press, 1997.

[77] B. Shahraray and D. C. Gibbon, "Automatic generation of pictorial transcripts of video programs," *Proc. SPIE 2417: Multimedia Computing and Networking*, Feb. 1995, pp. 512–518.

[78] B. Shahraray, "Scene change detection and content-based sampling of video sequences," *Proc. SPIE 2419: Digital Video Compression: Algorithms and Technologies*, Feb. 1995, pp. 2–13.

[79] M. Riley, A. Ljolje, D. Hindle, and F. Pereira, "The AT&T 60 000 word speech-to-text system," in *Proc. EuroSpeech'95*, pp. 207–210.

[80] A. G. Hauptmann and M. J. Witbrock, "Informedia news-on-demand: Using speech recognition to create a digital video library," in *Proc. AAAI Spring Symp. Intelligent Integration and Use of Text, Image, Video, and Audio Corpora*, Mar. 1997, pp. 120–126.

Richard V. Cox (Fellow, IEEE) received the B.S. degree from Rutgers—The State University, New Brunswick, NJ, and the M.A. and Ph.D. degrees from Princeton University, Princeton, NJ, all in electrical engineering.

He joined Bell Laboratories in 1979 and has worked in various aspects of speech and audio coding, speech privacy, digital signal processing, combined speech and channel coding for noisy channels, and real-time signal-processing implementations. In 1987, he became Supervisor of the Digital Principles Research Group. In 1992, he became Head of the Speech Coding Research Department. In AT&T Labs Research, he currently is Division Manager of the Speech Processing Software and Technology Research Department, with responsibility for speech and audio coding, text-to-speech synthesis, and human hearing research. He has been active in the creation of speech-coding standards for digital cellular telephony and the toll network. He was one of the creators of ITU-T Recommendation G.728 and was the Editor for ITU-T Recommendation G.723.1.

Dr. Cox is active in the IEEE Signal Processing Society. He is a past Chairman of its Speech Technical Committee, was a member of the Advisory Committee and Board of Governors for six years, and was Treasurer/Vice President of Finance. He currently is Vice President for Publications.

Barry G. Haskell (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1964, 1965, and 1968, respectively.

From 1964 to 1968, he was a Research Assistant in the University of California Electronics Research Laboratory, with one summer spent at the Lawrence Livermore National Laboratory. From 1968 to 1996, he was with AT&T Bell Laboratories, Holmdel, NJ. Since 1996, he has been with AT&T Labs, Middletown, NJ, where he presently is Division Manager of the Image Processing and Software Technology Research Department. He also has been an Adjunct Professor of Electrical Engineering at Rutgers—The State University, New Brunswick, NJ, City College of New York, and Columbia University, New York. Since 1984, he has been very active in the establishment of international video communications standards. These include International Telecommunications Union—Telecommunications Sector for video conferencing standards (H-series), ISO Joint Photographic Experts Group for still images, ISO Joint Bilevel Image Group for documents, and ISO Motion Picture Experts Group for digital television. His research interests include digital transmission and coding of images, video telephone, satellite television transmission, and medical imaging, as well as most other applications of digital image processing. He has published more than 60 papers on these subjects and has more than 40 patents either received or pending. He is the Coauthor of a number of books, most recently *Digital Video—An Introduction to MPEG-2* (New York: Chapman and Hall, 1997).

Dr. Haskell is a member of Phi Beta Kappa and Sigma Xi.

Yann LeCun (Member, IEEE) was born near Paris, France, in 1960. He received the Diplome d'Ingenieur from the Ecole Superieure d'Ingenieur en Electrotechnique et Electronique, Paris, in 1983 and the Ph.D. degree in computer science from the Universite Pierre et Marie Curie, Paris, in 1987.

During his doctoral studies, he proposed an early version of the back-propagation learning algorithm for neural networks. He joined the Department of Computer Science at the University of Toronto as a Research Associate. In 1988, he joined the Adaptive Systems Research Department, AT&T Bell Laboratories, Holmdel, NJ, where he worked on neural networks, machine learning, and handwriting recognition. Following AT&T's split in 1996, he became Department Head in the Image Processing Services Research Laboratory at AT&T Labs Research. He is a member of the editorial board of *Machine Learning Journal*. He is General Chair of the "Machines that Learn" workshop held every year since 1986 in Snowbird, UT. He has served as Program Cochair of IJCNN'89, INNC'90, and NIPS'90, '94, and '95. He has published more than 70 technical papers and book chapters on neural networks, machine learning, pattern recognition, handwriting recognition, document understanding, image processing, very-large-scale-integration design, and information theory. In addition to the above topics, his current interests include video-based user interfaces, image compression, and content-based indexing of multimedia material.

Dr. LeCun was an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS.

Behzad Shahraray (Member, IEEE) received the B.S. degree in electrical engineering from Arya-Mehr University, Tehran, Iran, in 1977 and the M.S.E. degree in electrical engineering, the M.S.E. degree in computer, information, and control engineering, and the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1978, 1983, and 1985, respectively.

He joined AT&T Bell Labs in 1985, where he was a Principal Investigator in the Machine Perception Research Department. Since 1996, he has been with AT&T Labs Research, where he currently heads the Multimedia Processing Research Department. His research interests are in the areas of computer vision, image processing, and multimedia computing and include feature extraction, motion estimation, video and multimedia indexing, and content-based multimedia information retrieval.

Dr. Shahraray is a member of the Association for Computing Machinery.

Lawrence Rabiner (Fellow, IEEE) was born in Brooklyn, NY, on September 28, 1943. He received the S.B. and S.M. degrees in 1964 and the Ph.D. degree in electrical engineering in 1967, all from the Massachusetts Institute of Technology, Cambridge.

From 1962 to 1964, he participated in the cooperative program in Electrical Engineering at AT&T Bell Laboratories, Whippany and Murray Hill, NJ. During this period, he worked on digital circuitry, military communications problems, and problems in binaural hearing. He joined AT&T Bell Labs in 1967 as a Member of Technical Staff. He became a Supervisor in 1972, Department Head in 1985, Director in 1990, and Functional Vice President in 1995. His research focused primarily on problems in speech processing and digital signal processing. Presently, he is Speech and Image Processing Services Research Vice President at AT&T Labs, Florham Park, NJ, and is engaged in managing research on speech and image processing and the associated hardware and software systems that support services based on these technologies. He is the Coauthor of a number of books, most recently *Fundamentals of Speech Recognition* (Englewood Cliffs, NJ: Prentice-Hall, 1993). He is a former Vice-President of the Acoustical Society of America.

Dr. Rabiner is a member of Eta Kappa Nu, Sigma Xi, Tau Beta Pi, the National Academy of Engineering, and the National Academy of Sciences. He is a Fellow of the Acoustical Society of America, Bell Laboratories, and AT&T. He is a former President of the IEEE Acoustics, Speech, and Signal Processing Society, a former editor of IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, and a former member of the Editorial Board of the PROCEEDINGS OF THE IEEE.