Multimedia Systems: An Overview

Borko Furht
Florida Atlantic University

Advances in distributed multimedia systems have begun to significantly affect the development of on-demand multimedia services. Researchers are working within established computer areas to transform existing technologies and develop new ones. The big picture shows multimedia as the merging of computing, communications, and broadcasting.

Multimedia systems combine a variety of information sources, such as voice, graphics, animation, images, audio, and full-motion video, into a wide range of applications. The big picture shows multimedia as the merging of three industries: computing, communication, and broadcasting.

Research and development efforts in multimedia computing fall into two groups. One group centers its efforts on the stand-alone multimedia workstation and associated software systems and tools, such as music composition, computer-aided learning, and interactive video. The other combines multimedia computing with distributed systems. This offers even greater promise. Potential new applications based on distributed multimedia systems include multimedia information systems, collaboration and conferencing systems, on-demand multimedia services, and distance learning.

The defining characteristic of multimedia systems is the incorporation of continuous media such as voice, video, and animation. Distributed multimedia systems require continuous data transfer over relatively long periods of time (for example, playout of a video stream from a remote camera), media synchronization, very large storage, and special indexing and retrieval techniques adapted to multimedia data types. The sidebar lists a number of acronyms relevant to multimedia systems.

Technical demands

A multimedia system can either store audio and video information and use it later in an application such as training, or transmit it live in real time. Live audio and video can be interactive, such as multimedia conferencing, or noninteractive, as in TV broadcasting. Similarly, stored still images can be used in an interactive mode (browsing and retrieval) or in a noninteractive mode (slide show).

The complexity of multimedia applications stresses all the components of a computer system. Multimedia requires great processing power to implement software codecs, multimedia file systems, and corresponding file formats. The architecture must provide high bus bandwidth and efficient I/O. A multimedia operating system should support new data types, real-time scheduling, and fast-interrupt processing. Storage and memory requirements include very high capacity, fast access times, and high transfer rates. New networks and protocols are necessary to provide the high bandwidth, low latency, and low jitter required for multimedia. We also need new object-oriented, user-friendly software development tools, as well as tools for retrieval and data management—important for large, heterogeneous, networked and distributed multimedia systems.

Abbreviations

ADPCM	adaptive differential pulse code modulation
ATM	asynchronous transfer mode
BER	bit error rate
B-ISDN	broadband integrated service digital network
CCITT	International Telegraph and Telephone Consultative Committee
Codec	coder/decoder
DCT	discrete cosine transform
DPCM	differential pulse code modulation
DSP	digital signal processor
DVI	digital video interactive
FDCT	forward discrete cosine transform
FDDI	fibre distributed data interface
IDCT	inverse discrete cosine transform
JPEG	Joint Photographic Expert Group
MMOS	multimedia operating system
MPEG	Moving Pictures Expert Group
NTSC	National Television Systems Committee
PAL	phase alternating line
PER	packet error rate
PTR	priority token ring
QOS	quality of service
RISC	reduced instruction set computer
STM	synchronous transfer mode

1070-986X/94/$4.00 © 1994 IEEE

Table 1. Storage requirements for various data types.

	Text	Image	Audio	Animation	Video
Object type	ASCII EBCDIC	Bitmapped graphics Still photos Faxes	Noncoded stream of digitized audio or voice	Synched image and stream at 15-19 frames/s	TV analog or digital image with synched streams at 24-30 frames/s
Size and bandwidth	2 KB per page	Sample: 64 KB per image Detailed (color) 7.5 MB per image	Voice/phone 8KHz/ 8 bits (mono) 6-44 KB/s Audio CD DA 44.1 kHz/ 16 bit 176 KB/s	2.5 MB/s for 320 × 640 × 16 pixels/frame (16 bit color) 16 frames/s	27.7 MB/s for 640 × 480 × 24 pixels per frame (24-bit color) 30 frames/s

KB= Kbytes MB=Mbytes

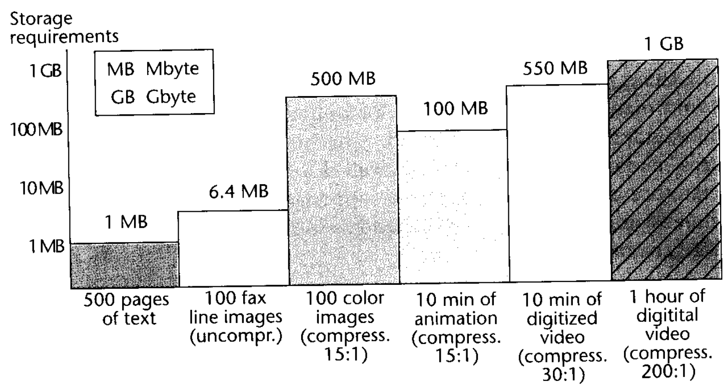


Figure 1. Storage requirements for a typical multimedia application with compressed images and video.

Researchers are working within established computer areas to transform existing technologies, or develop new technologies, for multimedia. This research involves fast processors, high-speed networks, large-capacity storage devices, new algorithms and data structures, video and audio compression algorithms, graphics systems, human-computer interfaces, real-time operating systems, object-oriented programming, information storage and retrieval, hypertext and hypermedia, languages for scripting, parallel processing methods, and complex architectures for distributed systems.

Multimedia compression

Compression techniques clearly play a crucial role in digital multimedia applications. Audio, image, and video signals produce a vast amount of data. Table 1 illustrates the mass storage requirements for various media types.

Present multimedia systems require data compression for three reasons: the large storage requirements of multimedia data, relatively slow storage devices that cannot play multimedia data (specifically video) in real time, and network bandwidth that does not allow real-time video data transmission.

For example, a single frame of a color video, with 620- × 560-pixel frames at 24 bits per pixel, would take up about 1 Mbyte. At a real-time rate of 30 frames per second, that equals 30 Mbytes for one second of video. A typical multimedia application might store more than 30 minutes of video, 2,000 images, and 40 minutes of stereo sound on each side of a laser disc. That application would require about 50 Gbytes of storage for video, 15 Gbytes for images, and 0.4 Gbytes for audio. That means a total of 65.4 Gbytes of storage on the whole disc.

Even if we had enough storage available, we wouldn't be able to play back the video in real time due to the insufficient bit rate of storage devices. The speed of a real-time storage device would need to be 30 Mbytes/s. However, today's CD-ROM technology provides a transfer rate of about 300 Kbytes/s. At the present state of storage device technology, the only solution is to compress the data before storage and decompress it before playback.

Modern image and video compression techniques reduce these tremendous storage requirements. Advanced techniques can compress a typical image at a ratio ranging from 10:1 to 50:1, achieving video compression up to 2,000:1.

Figure 1 illustrates storage requirements for a multimedia application consisting of various media types, compressing the images by a ratio of 15:1 and the video by factors of 30:1 and 200:1. The total storage requirement for this application becomes a little over 2 Gbytes, much more feasible than 225.5 Gbytes uncompressed.

Digital data compression relies on various computational algorithms, implemented either in software or in hardware. We can classify compression

Table 2. Multimedia compression standards.

Short name	Official name	Standards group	Compression ratios
JPEG	Digital compression and coding of continuous-tone still images	Joint Photographic Experts Group	15:1 (full color still-frame applications)
H.261 px64	Video coder/decoder for audio-visual services at px64 Kbps	Specialist Group on Coding for Visual Telephony	100:1 to 2000:1 (video-based tele-communications)
MPEG	Coding of moving pictures and associated audio	Moving Pictures Experts Group	200:1 Motion-intensive applications

techniques into lossless and lossy approaches.[1] Lossless techniques can recover the original representation perfectly. Lossy techniques recover the presentation with some loss of accuracy. The lossy techniques provide higher compression ratios, though, and therefore are applied more often in image and video compression than lossless techniques.

We can further divide the lossy techniques into prediction-, frequency-, and importance-based techniques. Predictive techniques (such as ADPCM) predict subsequent values by observing previous values. Frequency-oriented techniques apply the discrete cosine transform (DCT), related to fast Fourier transform. Importance-oriented techniques use other characteristics of images as the basis for compression; for example, the DVI technique employs color lookup tables and data filtering.

Hybrid compression techniques combine several approaches, such as DCT and vector quantization or differential pulse code modulation. Various groups have established standards for digital multimedia compression based on the existing JPEG, MPEG, and px64 standards, as Table 2 shows.

JPEG

Originally, JPEG targeted full-color still frame applications, achieving a 15:1 average compression ratio.[2,3] However, some real-time, full-motion video applications also use JPEG. The JPEG standard offers four modes of operation:

1. sequential DCT-based encoding, which encodes each image component in a single left-to-right, top-to-bottom scan;

2. progressive DCT-based encoding, which encodes the image in multiple scans in order to produce a quick, rough, decoded image when the transmission time is long;

3. lossless encoding, which encodes the image to guarantee an exact reproduction; and

4. hierarchical encoding, which encodes the image at multiple resolutions.

The JPEG algorithm decomposes the input image into 8×8 source blocks. It shifts the pixels, originally in the range 0 to 511, to the range of -128 to $+128$, then transforms them into the frequency domain using forward discrete cosine transform (FDCT). The transformed 64-point discrete signal is a function of two spatial dimensions, x and y. Its components are called spatial frequencies, or DCT coefficients.

For a typical 8×8 image block, most spatial frequencies have zero or near-zero values and need not be encoded. This is the foundation for data compression. In the next step, all 64 DCT coefficients are quantized with the 64-element quantization table specified by the application. Quantization reduces the amplitude of the coefficients that contribute little or nothing to the quality of the image, thus increasing the number of zero-value coefficients.

After quantization, the DCT coefficients are ordered into the "zig-zag" sequence shown in Figure 2a (see next page), because the low-frequency coefficients are more likely to be nonzero than the high-frequency coefficients (Figure 2b).

Finally, the last stage of JPEG is entropy coding. The JPEG standard specifies two entropy coding methods: Huffman coding and arithmetic coding. The technique provides additional compression by encoding the quantized DCT coefficients into a more compact form.[3]

MPEG

The MPEG standard is intended for compressing full-motion video.[4] It uses interframe compression, achieving compression ratios of up to 200:1 by storing only the differences between suc-

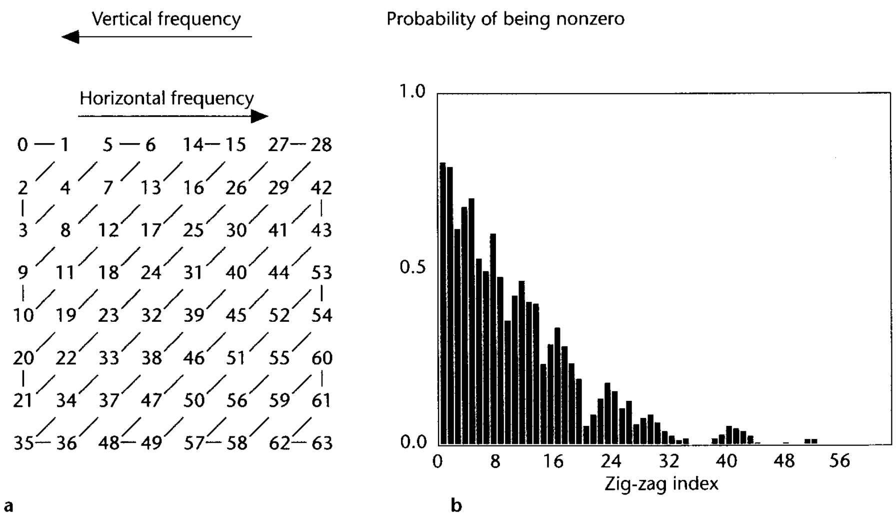


Figure 2. (a) Zig-zag ordering of DCT coefficients. (b) The probability of the coefficients being nonzero.[3]

cessive frames. MPEG specifications also include an algorithm for compressing audio data at ratios from 5:1 to 10:1.

MPEG codes frames in a sequence using three different algorithms, as Figure 3 shows. A DCT-based algorithm similar to JPEG first codes intraframes (I). To exploit temporal redundancy between frames, MPEG codes the remaining frames using two prediction techniques. One codes predicted frames (P) with forward predictive coding, where the actual frame is coded with reference to a past frame. The other codes interpolated, or bidirectional, frames (B) with bidirectionally predicted, interpolated coding, also called motion-compensated interpolation. Bidirectional prediction uses a past and a future frame to code current frames, providing the highest amount of compression.

The coding process for P and B frames includes a motion estimator that finds the best matching block common to the reference frames. The motion vector then specifies the distance between predicted and actual blocks. The difference, called the error term, is then encoded using the DCT-based transform coding.

The present standard, called MPEG-1, compresses 320 × 240 full-motion video in applications such as interactive multimedia and broadcast television. The minimum data rate required is 1.5 Mbps.

MPEG-2 will compress 720 × 480 full-motion video in broadcast television and video-on-demand applications. It will require a data rate in the range of 4 to 10 Mbps and will provide VCR-quality video.

Future broadcast television and video-on-demand services will use MPEG-3 to compress full-motion, HDTV-quality video. The projected required data rate is 5 to 20 Mbps.

Full-motion video applications, such as interactive multimedia and video telephony, that consist of small frames and require slow refreshing will use MPEG-4. Such applications will need a data rate of 9 to 40 Kbps.

px64

The H.261 standard, commonly called px64, achieves very high compression ratios for full-color, real-time motion video transmission. The algorithm combines intraframe and interframe coding to provide fast processing for on-the-fly video compression and decompression, optimized for applications such as video-based telecommunications. Because its applications usually are not motion-intensive, the algorithm uses limited motion search and estimation strategies to achieve higher compression ratios. For standard video communication images, px64 can achieve compression ratios of 100:1 to more than 2,000:1.

The standard covers the entire ISDN channel capacity (px64 Kbps, $p = 1, 2, \ldots, 30$). This increases the ISDN channel capacity from 64 Kbps to 2.048 Mbps. The video coding algorithm is intended for real-time communications requiring minimum delays. For $p = 1$ or 2, due to limited available bandwidth, this algorithm only implements desktop face-to-face visual communications (videophone). However, for p of 6 or higher, more complex pictures are transmitted, suiting the algorithm for videoconferencing.

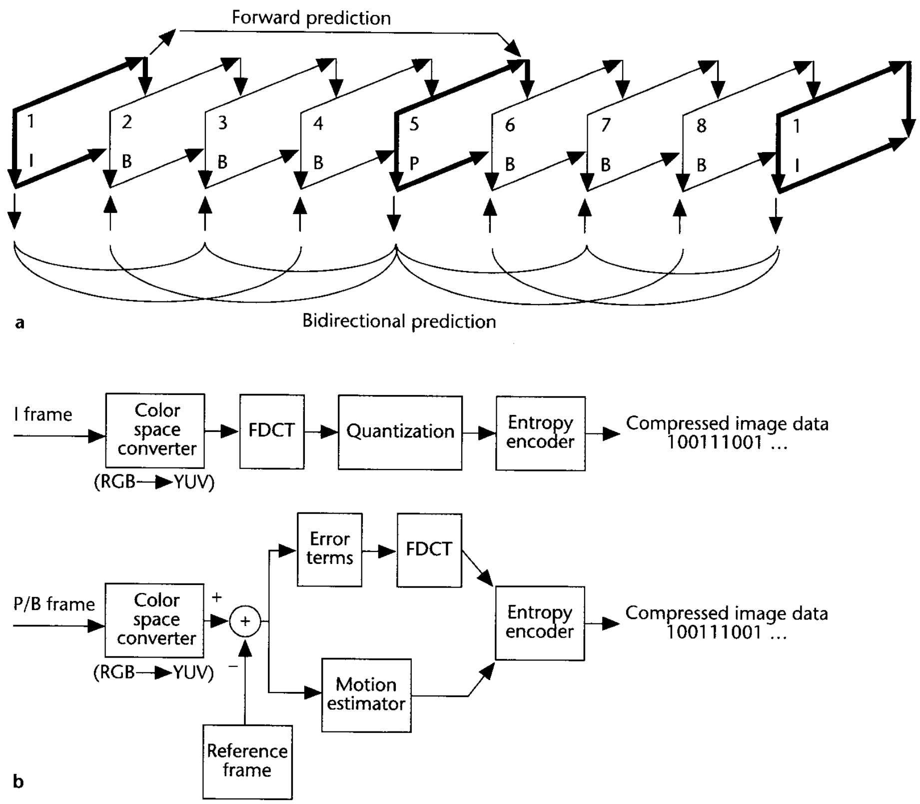


Figure 3. The MPEG compression algorithm, showing (a) a group of frames and (b) the MPEG coding procedure.

px64 operates with two picture formats adopted by the CCITT: the common intermediate format (CIF), and the quarter-CIF (QCIF).[5] The standard consists of a DCT-based compression algorithm, similar to JPEG, and a differential pulse code modulation (DPCM) algorithm with motion estimation.

Intraframe mode codes and quantizes frames using the DCT transform coding, then sends each to the video multiplex coder. The inverse quantizer and IDCT decompress the frames, then store them in the picture memory for interframe coding.

Interframe coding uses DPCM-based prediction to compare every macro block of the actual frame with the available macro blocks of the previous frame. The algorithm then creates the difference as error terms that are DCT-coded, quantized, and sent to the video multiplex coder along with the motion vector. The final stage uses variable word-length entropy coding (such as the Huffman coder) to produce more compact code.

Implementing compression algorithms

When implementing a compression/decompression algorithm, the key question is how to partition between hardware and software in order to maximize performance and minimize cost. Most implementations use specialized video processors and programmable digital signal processors (DSPs). However, powerful RISC processors are making software-only solutions feasible.

We can classify implementations of compression algorithms into three categories: (1) a hardwired approach that maximizes performance (for example, C cube), (2) a software solution that emphasizes flexibility with a general-purpose processor, and (3) a hybrid approach that uses specialized video processors.

AT&T took the hybrid approach, creating the AVP 4310E encoder and the AVP 4220D decoder for the px64 and MPEG standards.[6] The encoder accepts video input at 30 frames/s and outputs data at a selectable data rate from 40 Kbytes/s to 4 Mbytes/s. The hardware implements computationally intensive functions, such as motion estimation and Huffman coding. The user can program key parameters, such as frame rate, delay, bit-rate, and resolution. A programmable RISC processor implements less stable functions.

Table 3. Traditional communications versus multimedia communications.

Characteristics	Data Transfer	Multimedia transfer
Data rate	Low	High
Traffic pattern	Bursty	Stream-oriented highly bursty
Reliability requirements	No loss	Some loss
Latency requirements	None	Low, for example, 20 ms
Mode of communication	Point-to-point	Multipoint
Temporal relationship	None	Synchronized transmission

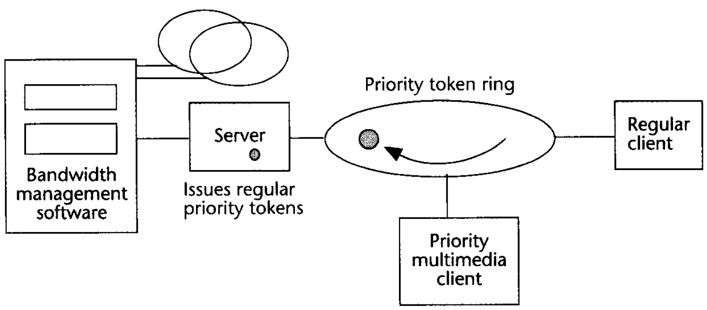


Figure 4. Priority token ring suits multimedia applications.

Researchers at UC Berkeley implemented a software MPEG encoder in C using X Windows and analyzed its performance on different computer platforms.[7] The results showed that current RISC workstations, such as the HP750, can decode a 320 × 240 video sequence at 10 to 15 frames/s, about half the rate of real-time performance. The group anticipated a new generation of workstations capable of real-time, software-only decoding of video signals.

Multimedia networking

Many applications, such as video mail, video conferencing, and collaborative work systems, require networked multimedia. In these applications, the multimedia objects are stored at a server and played back at the clients' sites. Such applications might require broadcasting multimedia data to various remote locations or accessing large depositories of multimedia sources.

Traditional LAN environments, in which data sources are locally available, cannot support access to remote multimedia data sources for a number of reasons. Table 3 contrasts traditional data transfer and multimedia transfer.

Multimedia networks require a very high transfer rate or bandwidth, even when the data is compressed. For example, an MPEG-1 session requires a bandwidth of about 1.5 Mbps. MPEG-2 through -4 will take 4 to 10 Mbps, while the projected required bandwidth for HDTV is 5 to 20 Mbps. Besides being high, the transfer rate must also be predictable.

Traditional networks are used to provide error-free transmission. However, most multimedia applications can tolerate errors in transmission due to corruption or packet loss without retransmission or correction. In some cases, to meet real-time delivery requirements or to achieve synchronization, some packets are even discarded. As a result, we can apply lightweight transmission protocols to multimedia networks. These protocols cannot accept retransmission, since that might introduce unacceptable delays.

Multimedia networks must provide the low latency required for interactive operation. Since multimedia data must be synchronized when it arrives at the destination site, networks should provide synchronized transmission with low jitter.

In multimedia networks, most communications are multipoint, as opposed to traditional point-to-point communication. For example, conferences involving more than two participants need to distribute information in different media to each participant. Conference networks use multicasting and bridging distribution methods. Multicasting replicates a single input signal and delivers it to multiple destinations. Bridging combines multiple input signals into one or more output signals, which it then delivers to the participants.[8]

Traditional networks do not suit multimedia. Ethernet provides only 10 Mbps, its access time is not bounded, and its latency and jitter are unpredictable. Token-ring networks provide 16 Mbps and are deterministic; from this point of view, they can handle multimedia. However, the predictable worst-case access latency can be very high.

An FDDI network provides 100 Mbps bandwidth, sufficient for multimedia. In the synchronized mode, FDDI has low access latency and low jitter. FDDI also guarantees a bounded access delay and a predictable average bandwidth for synchronous traffic. However, due to the high cost, FDDI networks are used primarily for backbone networks, rather than networks of workstations.

Less expensive alternatives include enhanced traditional networks. Fast Ethernet, for example, provides up to 100 Mbps bandwidth. Priority token ring is another system.

In priority token ring networks, the multimedia traffic is separated from regular traffic by priority (see Figure 4). The bandwidth manager plays a crucial role by tracking sessions, determining ratio priority, and registering multimedia sessions. Priority token ring (PTR) works on existing networks and does not require configuration control.

The admission control in PTR guarantees bandwidth to multimedia sessions; however, regular traffic can experience delays. For example, let's assume a priority token ring network at 16 Mbps that connects 32 nodes. When no priority scheme is set, each node gets an average of 0.5 Mbps of bandwidth. When half the bandwidth (8 Mbps) is dedicated to multimedia, the network can handle about 5 MPEG sessions (at 1.5 Mbps). In that case, the remaining 27 nodes can expect about 8 Mbps divided by 27, or 0.296 Mbps, about half of what they would get without priority enabled.

Crimmins[9] evaluated three priority ring schemes for their applicability to video conferencing applications: (1) equal priority for video and asynchronous packets, (2) permanent high priority for video packets and permanent low priority for asynchronous packets, and (3) time-adjusted high-priority for video packets (based on their ages) and permanent low priority for asynchronous packets.

The first scheme, which entails direct competition between video conference and asynchronous stations, achieves the lowest network delay for asynchronous traffic. However, it reduces the video conference quality. The second scheme, in which video conference stations have the permanent high priority, produces no degradation in conference quality, but increases the asynchronous network delay. Finally, the time-adjusted priority system provides a trade-off between first two schemes. The quality of video conferencing is better than in the first scheme, while the asynchronous network delays are shorter than in the second scheme.[9]

Present optical network technology can support the Broadband Integrated Services Digital Network (B-ISDN) standard, expected to become the key network for multimedia applications.[10] B-ISDN access can be basic or primary. Basic ISDN access supports 2B + D channels, where the transfer rate of a B channel is 64 Kbps, and that of a D channel is 16 Kbps. Primary ISDN access supports 23B + D in the US and 30B + D in Europe.

The two B channels of the ISDN basic access provide 2 × 64 Kbps, or 128 Kbps of composite bandwidth. Conferences can use part of this capacity for wideband speech, saving the remainder for purposes such as control, meeting data, and compressed video.[11]

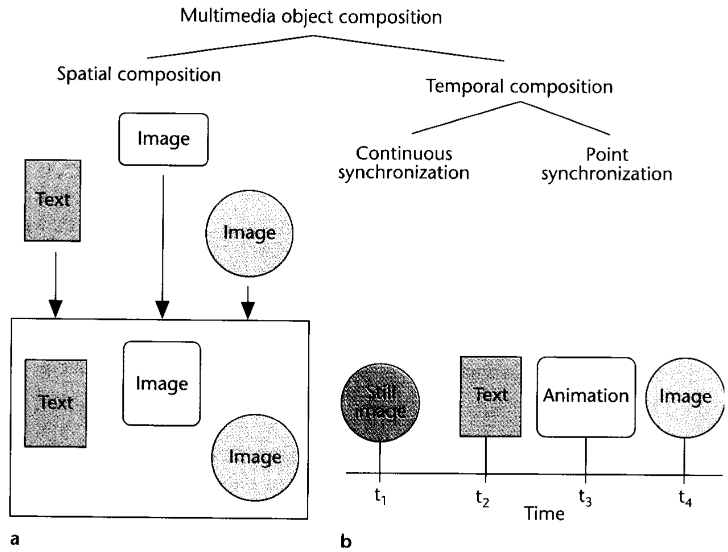


Figure 5. Multimedia objects composition.

Proposed B-ISDN networks are in either synchronous transfer mode (STM) or asynchronous transfer mode (ATM), to handle both constant and variable bit-rate traffic applications. STM provides fixed bandwidth channels, and therefore is not flexible enough to handle the different types of traffic typical in multimedia applications. On the other hand, ATM is suitable for multimedia traffic; it provides great flexibility in bandwidth allocation by assigning fixed length packets, called *cells*, to virtual connection. ATM can also increase the bandwidth efficiency by buffering and statistically multiplexing bursty traffic at the expense of cell delay and loss.[10]

Multimedia synchronization

Multimedia systems include multiple sources of various media either spatially or temporally to create composite multimedia documents. Spatial composition links various multimedia objects into a single entity (Figure 5a), dealing with object size, rotation, and placement within the entity. Temporal composition creates a multimedia presentation by arranging the multimedia objects according to temporal relationship (Figure 5b).[12]

We can divide temporal composition, or *synchronization*, into continuous and point synchronization. Continuous synchronization requires constant synchronization of lengthy events. An example of continuous synchronization is video telephony, where audio and video signals are created at a remote site, transmitted over the network, then synchronized continuously at the

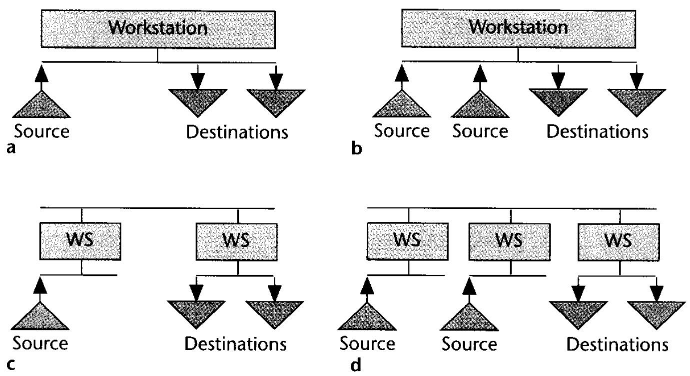


Figure 6. Location models for multimedia data: (a) local single source, (b) local multiple source, (c) distributed single source, and (d) distributed multiple source.

receiver site for playback. In point synchronization, a single point of one media block coincides with a single point of another media block. An example of point synchronization is a slide show with blocks of audio allotted to each slide.

Two further classes of synchronization are *serial* and *parallel* synchronization. Serial synchronization determines the rate at which events must occur within a single data stream (intramedia synchronization), while parallel synchronization determines the relative schedule of separate synchronization streams (intermedia synchronization).

Data location models

Responsibility for maintaining intermedia synchronization falls onto both the sources and destinations of data, but most techniques rely more on the destinations. One topology classification that can determine the required synchronization builds on data location models.[12,13] Figure 6 shows four data location models:

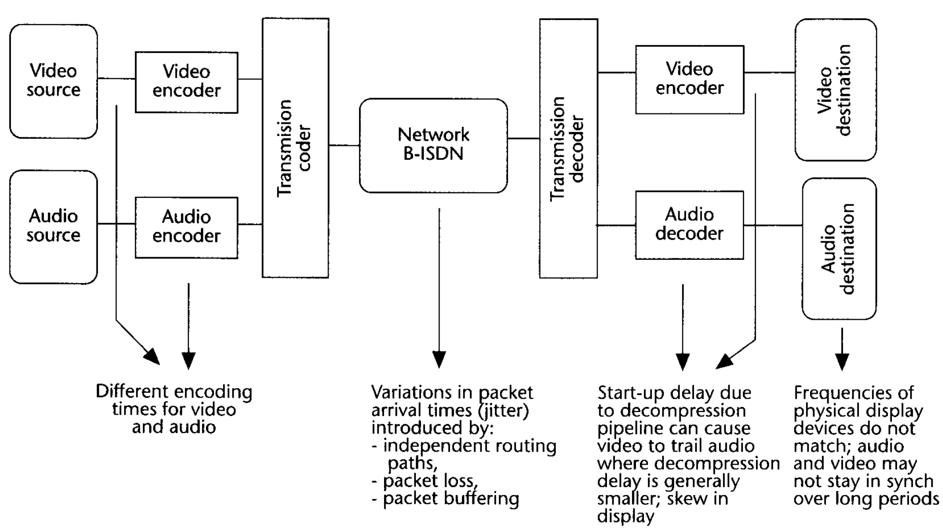
1. Local single source. A single source, such as a CD-ROM, distributes the media streams to the playback devices. As long as the devices maintain their playback speed, no synchronization technique is required.

2. Local multiple sources. More than one source distributes media streams to the playback devices. An example is a slide show played with music or an audio tape. Synchronization is required within the workstation.

3. Distributed single source. One source, such as a videotape, distributes media streams across a network to one or more nodes' playback devices; an example is cable TV. The technique requires no synchronization other than maintaining the speeds of the playback devices.

4. Distributed multiple sources. This is the most complex case, where more than one source distributes media streams to multiple playback devices on multiple nodes. This group further breaks down into multiple sources from one node distributed to another

Figure 7. Various causes of asynchrony in a video telephony system.[12]



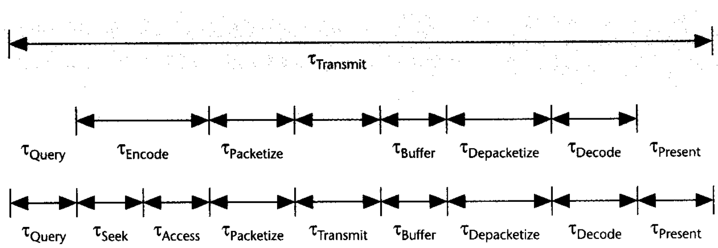


Figure 8. Definition of the end-to-end delay.

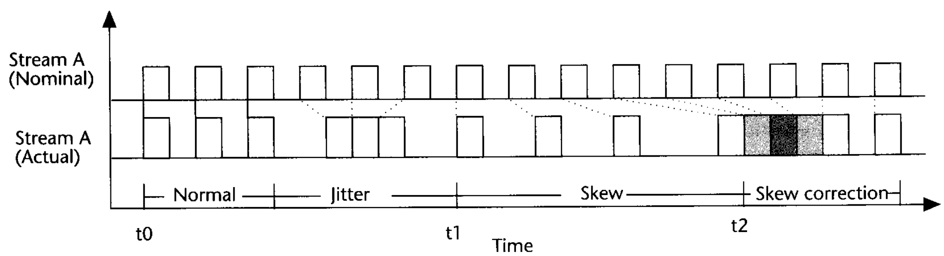


Figure 9. Nominal and actual streams, showing skew, jitter, utilization, and speed.

node (for example, a video call); multiple sources from two or more nodes distributed to another node; multiple sources from one node distributed to two or more nodes (for example, HDTV); and multiple sources from two or more nodes distributed to two or more nodes (for example, a group teleconference).

For the first two location models, local synchronization within the workstation suffices. However, the two cases with distributed sources require more complex synchronization algorithms to eliminate the various causes of asynchrony.[12] Figure 7 shows an example of video telephony, noting various places within the systems that contribute to asynchrony. The task of synchronization, whether implemented in the network or in the receiver, is to eliminate all the variations and delays incurred during the transmission of multiple media streams and to maintain synchronization among the media streams.

The end-to-end delay of a distributed multimedia system consists of all the delays created at the source site, network, and receiver site.[12] It differs slightly for real-time video and audio and for stored multimedia objects, as shown in Figure 8.

Quality of service

Implementing a synchronization algorithm for a specific application requires specifying the quality of service (QOS) for multimedia communications. The QOS is a set of parameters that includes speed ratio, utilization, average delay, jitter, bit error rate, and packet error rate.[12] Figure 9 illustrates how to calculate some of these parameters.

The speed ratio is the quotient of the actual presentation rate divided by the nominal presentation rate. For the interval $\{t_0, t_1\}$ in Figure 9, the speed ratio equals 6 over 6, or 1. For the interval $\{t_1, t_2\}$, the speed ratio equals 4 divided by 6, or 0.67.

The utilization ratio equals the actual presentation rate divided by the available delivery rate. Ideally, both the speed and utilization ratios equal 1. Duplication of frames will create a utilization ratio greater than 1, while frame dropping will cause it to be less than 1.

Skew is the average difference in presentation times between two synchronized objects over *n* synchronization points.[12] Duplicating a data frame causes the stream to retard in time (stream lag). Data losses resulting in playout gaps cause streams to advance in time (stream lead). In Figure 9, for the time interval $\{t_1, t_2\}$, the skew is 4 divided by 6.

Jitter is the instantaneous difference between two synchronized streams. Figure 9 shows how to correct jitter by dropping or duplicating frames. Figure 9 also shows how to correct skew by dropping frames. In this example, three shaded frames are dropped to reestablish synchronization.

Two more QOS parameters, the bit error rate (BER) and the packet error rate (PER), specify the

Table 4. Quality of service requirements.

	Video telephony	JPEG Video transmission
Speed ratio	1.0	1.0
Utilization	1.0	1.0
Average delay	0.25 s	0.2 s
Maximum jitter	10 ms	5 ms
Maximum BER	0.01	0.1
Maximum PER	0.001	0.01

required reliability of the network. Table 4 compares the quality of service required for video telephony and JPEG video transmission.

Single and multiple stream synchronization

Synchronization entails evaluating the temporal characteristics of the data streams to be synchronized and correcting all delays and other anomalies.[14]

To synchronize an event, first analyze the end-to-end delay, or latency. Then schedule a retrieval time that allows enough time before the deadline to allow for latency. For example, if the total latency time of retrieving a one-hour video is three minutes, and the customer ordered the video for 7 p.m., set the retrieval time, or packet production time, for 6:57 p.m. at the latest. Figure 10 shows timing of the single event synchronization.

We can extend single-event synchronization to single data stream synchronization, and further to the general case of multiple data streams synchronization.[14]

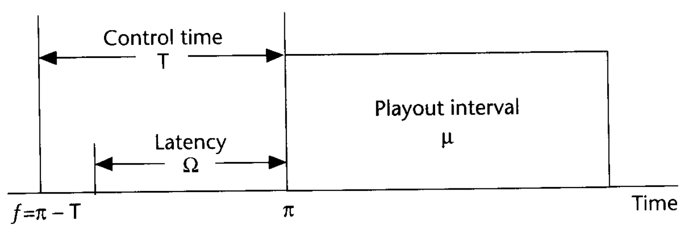


Figure 10. Timing for single-event synchronization.

The client site receives packets of data, created at a source and transmitted over the network, with a random distribution. We can reduce arriving packet delay variance using buffering or some other techniques. Then, synchronization consists of calculating the control time based on multiple deadlines, playout times, and latencies. In this case, time T represents the time required for buffering at the receiving site to smooth out variations in latencies.

We can extend synchronization of single streams of packetized audio and video to the general case of synchronizing multiple streams as well as nonstream data (for example, still images and text).[14] Then, various techniques determine the delay and buffering required to synchronize multiple events for given packet loss probabilities and network delay distributions.

Multimedia systems

Advances in several technologies are making multimedia systems technically and economically feasible. These advances include powerful workstations, high-capacity storage devices, high-speed networks, advances in image and video processing (such as animation and graphics), advances in audio processing (such as music synthesis and sound effects), speech processing (speaker recognition and text-to-speech conversion), and advanced still, video, audio, and speech compression algorithms.

A multimedia system consists of three key elements: multimedia hardware, operating system and graphical user interface, and multimedia software development and delivery tools (referred to as authoring tools). Since 1989, when the first multimedia systems were developed, it has been possible to differentiate the three generations of multimedia systems (see Table 5).

The first generation, based on Intel 80386 and Motorola 68030 processors, is characterized by bitmapped images and animation, JPEG video compression techniques, local area networks based on Ethernet and token ring, and hypermedia authoring tools. The second generation uses i80486 and MC68040 processors, moving and still images, 16-bit audio, JPEG and MPEG-1 video compression, FDDI networks, and object-oriented, multimedia authoring tools that incorporate text, graphics, animation, and sound.

We are presently at the transition stage from the second- to the third-generation systems, based on more powerful processors such as Pentium and PowerPC. The third generation will use full-motion, VCR-quality video, eventually moving to NTSC/PAL and HDTV. Compression algorithms will include MPEG-2, -3, and -4, and perhaps the wavelets method now in the research stage. The systems will use enhanced Ethernet, token ring, and FDDI networks, as well as new isochronous and ATM networks. The authoring tools will integrate object-oriented multimedia into the operating systems.[15]

Commercial approaches to designing multi-

Table 5. The three generations of multimedia systems.

	First generation 1989-91	Second generation 1992-94	Third generation 1995-96
Media	Text Black/white graphics Bit-mapped images Animation	Color bit-mapped images 16-bit audio Moving still images Full-motino video (15 frames/s)	Full-motion video (30 frames/s) (NTSC/PAL and HDTV quality)
Authoring capability	Hypertext Hypermedia	Object-oriented multimedia with text, graphics, sound, animation, still images, and full-motion video	Integration of object-oriented multimedia with operating systems
Video compression technology	DCT JPEG	Motion JPEG MPEG-1	MPEG-2,3,4 Wavelets
Base platform	25 MHz 386 (68030) 2MB DRAM 40 MB Hard disk VGA Color (680 × 740) 500 MB CD-ROM (100Kb/s)	50 MHz 486 (68040) 8-16 MB DRAM 240 MB Hard disk 1-2 1.5 MB Floppies VGA with 256 colors (1024 × 768) 500 MB CD-ROM (150 KB/s)	50-100 MHz Pentium (PowerPC) 16-32 MB DRAM 1-2 600 MB Hard disk 20-30 MB Floppies SVGA (1280 × 960) 50 MB Writable CD-ROM (300 KB/s)
Operating system	DOS	DOS 5 Windows 3.x OS/2 Presentation Manager	Windows NT Pink (IBM/Apple)
Delivery mode	720 KB Diskette 1.5 MB Laserdisk (R/O) 128 MB CD-ROM (R/O)	500 MB CD-ROM (R/O)	500 MB WORM 128-500 MB Magnetoptic (R/W)
Local area network	Ethernet (10 Mb/s) Token ring (16 Mb/s)	FDDI (100 Mb/s)	Ethernet, Token ring (100Mb/s) FDDI (500 MB/s) Isochronous networks ATM

MB=Mbytes KB=Kbytes Mb=Mbits

media systems and related tools include IBM's Ultimedia System, Microsoft's Windows Multimedia Extensions, Apple's QuickTime, and others.

Applications

Multimedia systems suggest a wide variety of potential applications. Three important applications already in use are multimedia mailing systems, collaborative work systems, and multimedia conferencing systems.

Multimedia mailing systems

Multimedia mailing systems are more sophisticated than standard electronic mailing systems. They implement multiple applications, such as multimedia editing and voice mail, and require higher transmission rates than text-only systems. Related projects include ARPA-sponsored research on multimedia mail in packet networks; the Modular Integrated Communications Environment (MICE) of Bellcore; and Etherphone, the voice-mailing system developed at Xerox PARC.[16]

The Etherphone system provides a variety of telephony and document applications within a distributed computing environment. The original system consists of a telephone/speakerphone instrument connected to the network by a standard telephone line. It can place and receive telephone calls, keep private telephone directories, and maintain a database of voice messages. The Macaw extension adds video capabilities. The Phoenix extension replaces the specialized telephones with the audio capabilities built into a Sun Sparcstation, which can digitize audio signals at 64 Kbps.

Collaborative work systems

Collaborative work systems allow group members to discuss a problem and actually create something together. During a meeting, users can view, discuss, and modify multimedia documents. The CES system, developed at the Massachusetts Institute of Technology, allows coauthors to work asynchronously on a shared document.

Colab is another system, developed by Xerox PARC, that provides real-time services. Colab allows two to six people to collaborate using personal computers interconnected over a LAN.[17]

Multimedia conferencing

Multimedia conferencing systems enable a number of participants to exchange various multimedia information via voice and data networks. Each participant has a multimedia workstation, linked to the other workstations over high-speed networks. Each participant can send and receive video, audio, and data, and can perform certain collaborative activities. The multimedia conference uses the concept of the *shared virtual workspace*, which describes the part of the display replicated at every workstation.

The biggest performance challenge in multimedia conferencing occurs when conference participants continuously transmit video and voice streams. Current research focuses on mixing these streams together to form a composite stream consisting of video and audio streams. Ramanathan et al. proposed techniques and protocols for media mixing and an optimal communication architecture for multimedia conferencing.[18]

Multimedia conferencing systems must provide a number of functions, such as multiple-call setup, conference status transmission, real-time control of audio and video, dynamic allocation of network resources, multiport data transfer, synchronization of shared workspace, and graceful degradation under fault conditions.

Prototype multimedia conferencing systems include MMConf, implemented in the Diamond system;[19] RTCal (Real Time Calculator), developed at MIT;[19] Rapport, developed by AT&T;[8] the European collaborative projects MIAC (Multipoint Interactive Audiovisual Communications) and MIAS (Multipoint Interactive Audiovisual System);[11] and IBM's Person-to-Person.

Research directions

Research and development in high-speed networks will soon provide the bandwidth needed for distributed multimedia applications. Therefore, I envision tremendous growth in distributed multimedia systems and their applications. One of the major projects in high-speed networks is Blanca, part of the Gigabit Testbed Initiative sponsored by the Corporation for National Research Initiatives and supported by the National Science Foundation. This project aims to build two wide-area, ATM-based testbeds—Xunet 2 at 45 Mbps and the 622-Mbps Xunet 3—and to demonstrate on them several multimedia applications that require high-speed networking.

Another nationwide initiative supported by the NSF is the development of digital multimedia libraries. In this area, a working group for the Electronic Library of the Future recently convened to conceive and develop the emerging "library without walls." The first project involves the Library of Congress. By the year 2000, the project hopes to have the core of the library's collection on-line and accessible over networks.

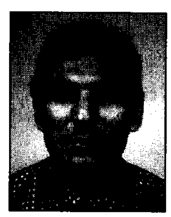
Advances in distributed multimedia systems have begun to significantly affect the development of on-demand multimedia services, such as interactive entertainment, video news distribution, video rental services, and digital multimedia libraries. Various companies realized that fiber-optic networks, coupled with improved computing and compression techniques, would soon be capable of delivering digital movies. Over the past year, a number of alliances have formed between entertainment, cable, phone, and computer companies, with the main focus on video-on-demand applications. Examples of these alliances are Bell Atlantic and Tele-Communications Inc. in the area of entertainment; Nynex and Dow Jones in video news distribution; IBM and Blockbuster in video rental services; and IBM, Apple, Motorola, and Scientific Atlanta in interactive television. Time Warner Cable is presently building a prototype interactive television system in Orlando, expected to become operational in 1994, which will serve about 4,000 customers with applications such as movies on demand, interactive games, home shopping, and distance learning.

Many challenging problems remain to be researched and resolved for the further growth of multimedia systems. Multimedia applications make enormous demands on computer hardware and software resources. Therefore, one of the ongoing demands is to develop more powerful multimedia workstations. Multimedia workstations will also need multimedia operating systems (MMOSs) and advanced multimedia user interfaces. An MMOS should handle continuous media by providing preemptive multitasking, easy expandability, format-independent data access, and support for real-time scheduling. It should be object-oriented and capable of synchronizing data streams. Users expect multimedia data to be instantly available, so the user interface must be highly sophisticated and intuitive. Integrating the user interface at the operating system level could eliminate many problems for application software developers.

Other research challenges include developing new real-time compression algorithms (perhaps based on wavelets), large storage devices, and multimedia data management systems. The constant challenge is further refinement of high-speed, deterministic networks with low latency and low jitter, as well as research in new multimedia synchronization algorithms. **MM**

References

1. E.A. Fox, "Advances in Interactive Digital Multimedia Systems," *Computer*, Vol. 24, No. 10, Oct. 1991, pp. 9-21.
2. G. Wallace, "The JPEG Still Picture Compression Standard," *Comm. ACM*, Vol. 34, No. 4, Apr. 1991, pp. 30-44.
3. W.B. Pennenbaker and J.L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.
4. D. LeGall, "MPEG: A Video Compression Standard for Multimedia Applications," *Comm. ACM*, Vol. 34, No. 4, Apr. 1991, pp. 45-68.
5. M. Liou, "Overview of the px64 Kbit/s Video Coding Standard," *Comm. ACM*, Vol. 34, No. 4, Apr. 1991, pp. 59-63.
6. B.D. Acidand et al., "A Video-Codec Chip Set for Multimedia Applications," *AT&T Tech. J.*, Vol. 72, No. 1, Jan./Feb. 1993, pp. 50-65.
7. K. Patel, B.C. Smith, and L.A. Rowe, "Performance of a Software MPEG Video Decoder," *Proc. ACM Multimedia 93*, ACM Press, New York, 1993, pp. 75-82.
8. S.R. Ahuja and J.R. Ensor, "Coordination and Control of Multimedia Conferencing," *IEEE Comm.*, Vol. 30, No. 5, May 1992, pp. 38-43.
9. S.M. Crimmins, "Analysis of Video Conferencing on a Token Ring Local Area Network," *Proc. ACM Multimedia 93*, ACM Press, New York, 1993, pp. 301-310.
10. G.M. Woodruff and R. Kositpaiboon, "Multimedia Traffic Management Principles for Guaranteed ATM Network Performance," *IEEE J. on Selected Areas in Comm.*, Vol. 8, No. 3, Apr. 1990, pp. 437-446.
11. W.J. Clark, "Multiport Multimedia Conferencing," *IEEE Comm.*, Vol. 30, No. 5, May 1992, pp. 44-50.
12. T.D.C. Little and A. Ghafoor, "Network Considerations for Distributed Multimedia Object Composition and Communication," *IEEE Network*, Vol. 4, No. 6, Nov. 1990, pp. 32-49.
13. D.C.A. Bulterman and R. van Liere, "Multimedia Synchronization and Unix," *Proc. 2nd Int'l Workshop on Network and Operating System Support for Digital Audio and Video*, Heidelberg, Germany, Nov. 1991, pp.108-119.
14. T.D.C. Little and A. Ghafoor, "Multimedia Synchronization Protocols for Broadband Integrated Services," *IEEE J. on Selected Areas in Comm.*, Vol. 9, No. 9, Dec. 1991, pp. 1,368-1,382.
15. B. Cole, "Interactive Multimedia: The Technology Framework," *IEEE Spectrum*, Vol. 30, No. 3, Mar. 1993, pp. 32-39.
16. H.M. Vin et al., "Multimedia Conferencing in the Etherphone Environment," *Computer*, Vol. 24, No. 10, Oct. 1991, pp. 69-79.
17. W.E. Kackay and G. Davenport, "Virtual Video Editing in Interactive Multimedia Applications," *Comm. ACM*, Vol. 32, No. 7, July 1989, pp. 802-810.
18. S. Ramanathan et al., "Optimal Communication Architecture for Multimedia Conferencing in Distributed Systems," Tech. Report No. CS91-213, Univ. of California, San Diego, Computer Science and Eng. Dept., Oct. 1991.
19. L.O. Barbora and N.D. Georganas, "Multimedia Services and Applications," *European Trans. on Telecommunications*, Vol. 2, No. 1, Jan./Feb. 1991, pp. 5-19.



Borko Furht is a professor of computer science and engineering at Florida Atlantic University and founder of its multimedia laboratory, funded by the NSF. His research is in multimedia, real-time systems, and advanced computer architecture. Furht holds a PhD in electrical and computer engineering and is a senior member of the IEEE.

Readers can contact Furht at the Department of Computer Science and Engineering, Florida Atlantic University, Boca Raton, FL 33431, e-mail borko@cse.fau.edu.