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

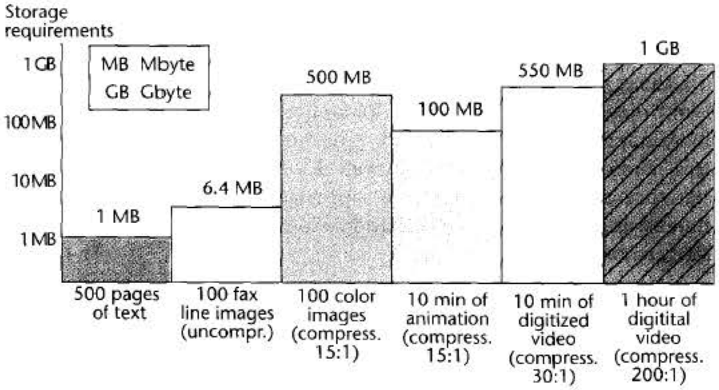

Figure 1. Storage requirements for a typical multimedia application with compressed images and video.

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-

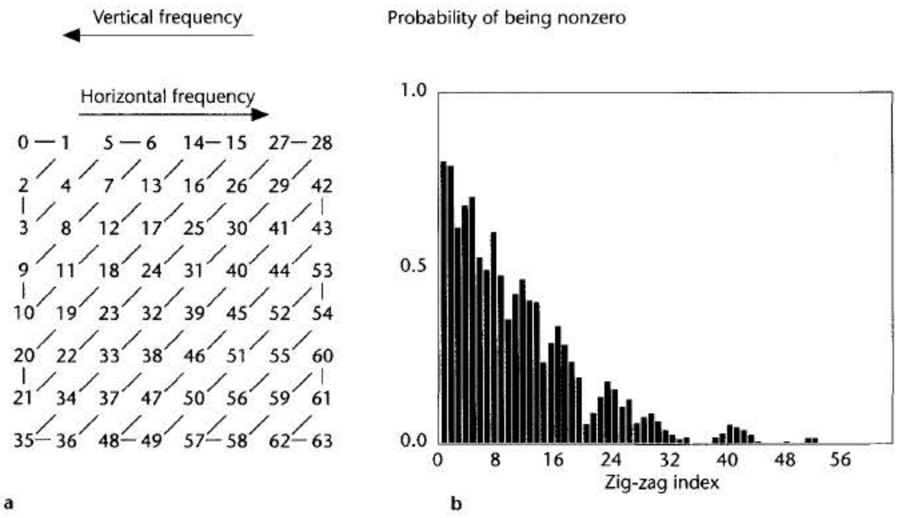

Figure 2. (a) Zig-zag ordering of DCT coefficients. (b) The probability of the coefficients being nonzero.[3]

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.

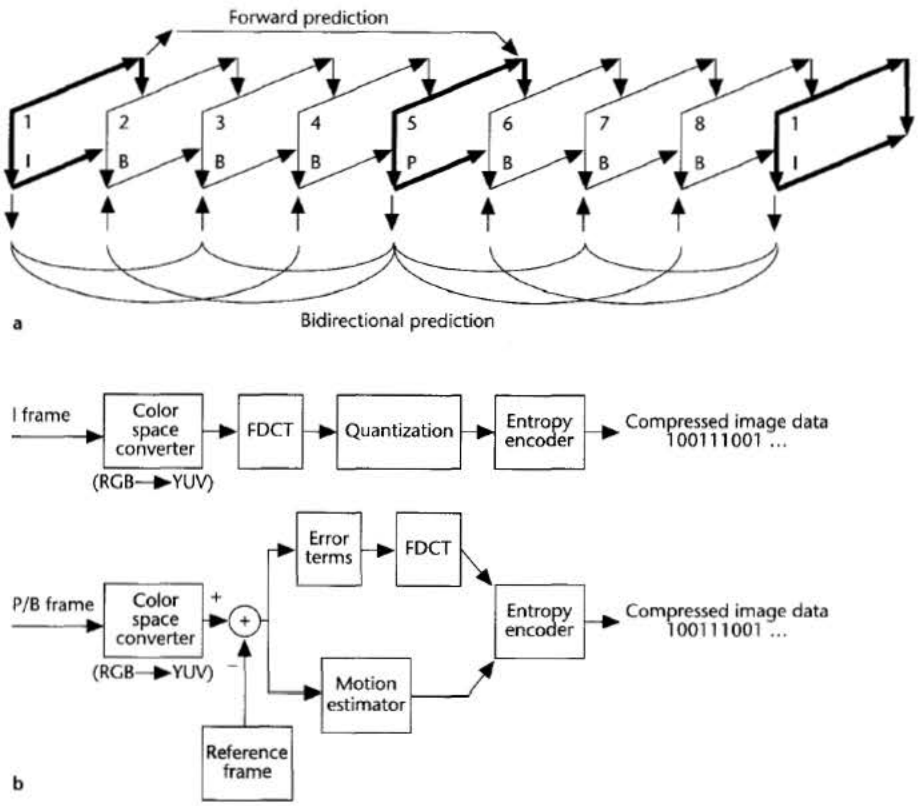

Figure 3. The MPEG compression algorithm, showing (a) a group of frames and (b) the MPEG coding procedure.

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.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**

fastcase®
Smarter legal research.